

For Users of Heath/Zenith Computers

SEXTANT

Issue No. 26

January-February 1987

\$2.95

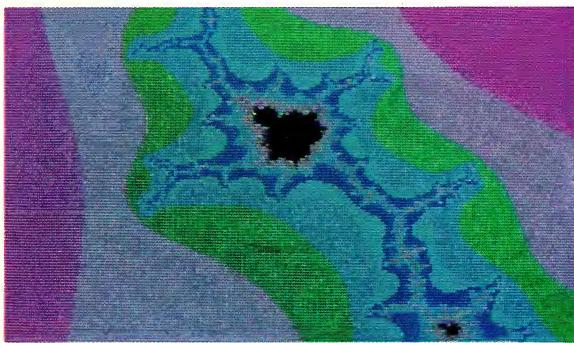


**The Programmer's
Utility Pack:
Useful Tools for
the '100 and '150**

**CHUGCON 86:
Old Friends and New**

**Four Upgrades
for the '89**

**Writing Hardware-
Independent Terminal
Emulators for
CP/M and MS-DOS**



6

In our July-August issue, Ed Byrnes presented short BASIC programs to create intricate images on the screen of the Z100 and Z150. We invited *Sextant* readers to explore this complex frontier of computer graphics and submit their results to us. Here are some of the fractals they came up with, and two more programs you can experiment with on your computer.



70

CHUGCON 86 gave us a chance to look at vendor support for Heath/Zenith users. We found vendors who were making their debuts in the Heath/Zenith market, others who were back for the fifth time, and many with new ideas and new products.

SEXTANT

Features

Winning Fractals

Edward A. Byrnes

6

Zenith: Planning for the Future

Wayne Rash, Jr.

13

Z100 Notebook

William N. Locke

17

We take a look at two useful, and under-used, MS-DOS utilities. We also consider ZP/SIM—which lets you run software under MS-DOS that was written for CP/M. The Z100 Code Corner offers a program to store the contents of the screen in a disk file.

The Programmer's Utility Pack: Useful Tools for the '100 and '150

William M. Adney

25

Zenith Data Systems and Heath Company offer about two dozen computer models capable of running MS-DOS. If you want to do more than run commercial software on them, you should consider spending \$225 on a PUP for your machine. Here's a description of what you get for the money.

The Eight-Bit World

Walter J. Janowski

33

Here's an 8-bit perspective on HUGCON 5 and HDOS 3. We consider a basic 8-bit survival system, and some vendors that can help.

Issue #26

January-February 1987

Four Upgrades for the '89

William Clarkson

Here are the adventures of an H89 user who decided to add a RAM disk and a parallel printer interface. He also added some more memory and doubled the clock speed. Perhaps you can profit from what he learned from his venture into systems integration.

C Notes

Joseph Katz

Here's a look at version 4 of Microsoft's C compiler—the CodeView debugger is a significant addition. We also check out screen generators and Dan Bricklin's Demo Program.

Writing Hardware-Independent Terminal Emulators for CP/M and MS-DOS

William S. Hall

It's possible to write a program to turn your computer into a dumb terminal even if you don't have detailed information about your system's hardware. Here are three examples. The first two run under Z-DOS or MS-DOS on the H/Z100 series computers. The other runs under CP/M on an H8, H/Z89, or H/Z100.

CHUGCON 86: Old Friends and New

Victoria Saxon

37

49

56

70



13

In November, Zenith Data Systems unveiled the Z386 shown above. There isn't any IBM computer for it to be compatible with. But ZDS has a strategy for compatibility with future IBM announcements—the same strategy embodied in the Z150 and Z200 line. Here's a look at that strategy. The flexibility for future upgrades goes in before the name goes on.

Departments

The Editorial Eye	2
Letters	3
Z100 Notebook	17
The Eight-Bit World	33
C Notes	76
Index to Advertisers	62
Classified Ads	76

The Editorial Eye

I noted last issue that Compaq Computer Corp. had unveiled its Deskpro 386 without waiting for IBM to announce a product designed around Intel's 80386 32-bit microprocessor. In November, Zenith Data Systems announced its 80386-based Z386 for delivery in 1987.

A couple of weeks before that announcement, the whole *Sextant* crew turned out for CHUGCON 86, which was held only a few miles from our office. (See page 70 for more on the conference.)

Members of the Capital Heath Users' Group had put together a good program for the weekend. I was especially interested in three tutorials on the 80286-based Z248 presented by John Guenther of Zenith Data Systems.

The Z248 is the 8-MHz member of the IBM-PC-AT-compatible Z200 line described in our March-April 1986 issue ("A Quick Look at the Z200," by Robert J. Gray). Its initial production was snatched up by various agencies of the Department of Defense under a contract first thought to be worth about \$242 million to ZDS. Guenther mentioned at CHUGCON 86 that the Navy had already ordered all the Z248s allocated to it through 1987. He put the Department of Defense backlog level at 31,000 units.

Despite the large military demand for the Z248, ZDS has set a goal of producing 1,500 to 2,000 Z248s per month for the commercial marketplace. I was considering getting one. But based on what I heard at CHUGCON 86, I ended up buying a Z110 for \$450.

That seemed a lot more appropriate for our immediate need of some additional capability for handling short pieces of text—letters and items for *Buss*. It would be nice to have some additional speed in processing long *Sextant* articles, but we'd already addressed that issue to a significant degree. Adding Software Wizardry's Wildfire to our Z151 provided a speed increase of about 60% overall—with individual jobs running from 10% to 100% faster.

Right now, the operating system that makes the best use of the Z248 is Xenix. John Guenther characterized it as a multi-user operating system that's disk-intensive and programmer-friendly—by contrast to MS-DOS, a single-user operating system which he called screen-intensive and user-friendly. Xenix costs about \$1,100 for a complete package taking up 11 megabytes of hard-disk storage. Guenther said that Xenix on a Z248 supports four simultaneous users well—and 16, not so well. (Sixteen users can share an inventory-control system, but you wouldn't want more than four users if the system is used for doing text processing or spreadsheets.)

Guenther's talk reinforced my preconception that Xenix would be of no more use to *Sextant* Publishing Co. than COBOL. I guess we'll wait for a version of MS-DOS that takes advantage of the strengths of the Z248 or Z386.

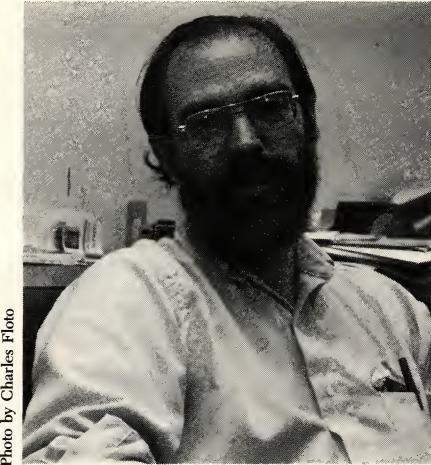


Photo by Charles Floto

SEXTANT®

Charles Floto

Editor
Publisher

John Gill
Assistant Publisher

Elizabeth Saxon-Giles
Managing Editor

John Walker
Technical Editor

Dana F. Lombardy
Advertising Manager

Gabriele Burkhard
Circulation Assistant
Production Assistant

Kieran O'Leary
Retail Sales Manager
Production Assistant

Beverly Voigt
Editorial Assistant
Production Assistant

Deborah L. Peel
Office Assistant

Edward A. Byrnes
Contributor

Walter J. Janowski
Contributor

Joseph Katz
Contributor

William N. Locke
Contributor

Subscription Service • 202/544-0900
Advertising Inquiries • 202/544-2868
Authors' Inquiries • 202/544-2867
Charles Floto • 202/544-0484

Sextant (ISSN 0731-2180) is published bimonthly by *Sextant* Publishing Co., 716 E Street, S.E., Washington, DC 20003. Second-class postage paid at Washington, DC and at additional mailing offices. POSTMASTER: Send address changes to *Sextant*, 716 E Street, S.E., Washington, DC 20003.

Copyright© 1986 by *Sextant* Publishing Co. All rights reserved. No part of this publication may be reprinted or otherwise reproduced without written permission from the publisher. Printed in the U.S.A. by The William Byrd Press, Richmond, Virginia 23261.

Subscription rates: 6 issues (1 year) in U.S.—\$14.97; in Canada and Mexico—\$17.25; overseas—\$21.00. Payment accepted by check in U.S. dollars payable on a U.S. bank, by U.S. and international postal money order, and by Visa and MasterCard. Please allow six to eight weeks for delivery.

"Heath" and "Heathkit" are trademarks of the Heath Company. "CP/M" is a trademark of Digital Research. "Unix" is a trademark of AT&T Bell Labs.

Letters

A C compiler for the '100

I read with interest the comments by Joseph Katz on the Computer Innovations' Optimizing C86 compiler ("C Notes," September-October 1986). He states that he knows of no C compiler package with H/Z100-specific libraries. He has apparently never discovered DeSmet C88—a fine C compiler available for about a quarter of the price of CI C86—which has a complete source code library of console I/O functions for the Z100, as well as ANSI and a number of other popular console/terminal types.

The C88 package includes a fast, easy-to-use full-screen editor and an optional (extra cost) symbolic debugger which are supplied configured for the PC compatibles. By using a simple process of relinking the supplied files, as described in the manual, you recreate both programs ready to run on a Z100.

If you write a C program containing these full-screen I/O functions, and want to run it on a real PC, then all you do is link it with the unmodified (PC-style) library. If you want to run it on a Z100, link it with your reconfigured library instead. If you want to run it on both machines, a bit of clever function renaming and programming is required.

Programs that don't need the full-screen I/O run on both machines without doing anything special. Numeric coprocessor (8087) support is fully supported by linking with a special library supplied with the package.

My guess is that the DeSmet C88 compiler is a far better buy for the Z100 owner than *any* C compiler costing over \$100. The CI C86 is way out of my price range, thank you.

Morrin E. Hazel, Jr.
Saugus, MA

DeSmet C88 (\$109, plus \$50 for optional D88 debugger) is available from C Ware Corporation, 505 W. Olive, Suite 767, Sunnyvale, CA 94086, 408/720-9696.

Saving fractal parameters

Edward Byrnes' BASIC fractals is sure an interesting program. [See "BASIC Fractals" in *Sextant* #23, July-August 1986.] It is, however, *slow*, as Mr. Byrnes well knew. Without his provision for storing a picture in progress and coming back to it later, the program would be useless because it would tie up your computer for days. My Z100 has been grinding away on one picture for 27 hours, total, and still isn't half finished. Very well, a provision has been made for calling a halt now and then to get some work done, and it must be used. However, as written, the pro-

gram calls up the various necessary parameters, displays them on the screen and then asks us to write them down on a scrap of paper so we can re-enter them by hand when we wish to resume. Write them down on paper? Re-enter them by hand? In this, the age of the computer, and using one of the durn things to boot,

we have to resort to handwritten notes on scratch paper? Come, come, now. This will never do!

[Listings A and B provide] patches to Mr. Byrnes' program that store the parameters of an unfinished picture in a file named **FRAC.DAT**. This is done automatically upon each work termination.

GW-BASIC VERSION FRACTALS by E. BYRNES... PATCH by B. NABUR

This version has been renumbered. Lines 50, 60, and 70 are new and to be inserted directly after line 40 in the original program. Line 160 (old line 120) has had its GOSUB renumbered. Same for line 190 (old line 150). Lines 370 on are new and to be inserted following the old line 370.

```
10 KEY OFF: CLS: PRINT TAB(30) "FRACTAL IMAGE GENERATOR"
20 ON ERROR GOTO 370: FILES"???????.PIC"
30 PRINT "The picture files above are currently on this drive."
40 PRINT: INPUT "The name of this fractal is (do not use . or extension)";A$
50 PRINT: PRINT: PRINT "Resume old picture? <Y/N> "
60 Q$=INPUT$(1)
70 IF Q$="Y" OR Q$="Y" GOTO 380
80 PRINT: PRINT
90 INPUT "Lambda and side":ACORNER,BCORNER,SIDE: GAPH=SIDE/200: GAPPX=SIDE/320
100 INPUT "Start at row: (row #1 for new image)":M1
110 PRINT: PRINT "BLOAD partial image? <Y/N> "; ZZ$=INPUT$(1): SCREEN 1: COLOR 0,
2
120 IF ZZ$="Y" OR ZZ$="Y" THEN CLS: GOSUB 350 ELSE CLS: GOTO 150
130 FOR M=M1 TO 200: FOR N=1 TO 319
140 AC=(N*GAPPX)+ACORNER: BC=(M*GAPH)+BCORNER: GOSUB 200
150 AZ=0: BZ=0
160 ZZ$=INKEY$: IF ZZ$="Z" THEN GOSUB 500: END
170 IF SIZEQ>4 THEN PSET(N,M),V
180 COUNT=0: SIZEQ=0
190 NEXT: NEXT: LOCATE1,1,1: GOSUB 500: END
200 AX=AZ^2-BZ^2+AC: BX=2*AZ*BZ+BC: AZ=AX: BZ=BX: SIZEQ=AX^2+BX^2
210 IF SIZEQ>4 THEN GOSUB 230: RETURN
220 COUNT=COUNT+1: IF COUNT>100 THEN 200 ELSE RETURN
230 IF COUNT>50 THEN V=1
240 IF COUNT<25 AND COUNT<=50 THEN V=2
250 IF COUNT>15 AND COUNT<=25 THEN V=3
260 IF COUNT>8 AND COUNT<=15 THEN V=1
270 IF COUNT>5 AND COUNT<=8 THEN V=2
280 IF COUNT>2 AND COUNT<=5 THEN V=3
290 IF COUNT>0 AND COUNT<=2 THEN V=1
300 RETURN
310 DEF SEG=&HB800: BSAVE A$+".PIC",0,64000!
320 PRINT "NAME: ";A$;"": PRINT "LAMBDA: ";ACORNER;"";BCORNER
330 PRINT "SIDE: ";SIDE: PRINT "ROW: ";M: PRINT "(write): PRINT "this": PRINT "down
n!""
340 RETURN
350 DEF SEG=&HB800: BLOAD A$+".PIC",0
360 RETURN
370 IF ERR=53 THEN PRINT "No picture files found.": RESUME 30
380 OPEN "1",#1,"FRAC.DAT"
390 IF EOF(1) GOTO 480
400 INPUT#1,AA$
410 INPUT#1,ACORNER
420 INPUT#1,BCORNER
430 INPUT#1,SIDE
440 INPUT#1,MM
450 IF A$<>AA$ GOTO 390
460 M1=MM
470 SCREEN 1: COLOR 0,2: CLOSE: CLS: GAPPX=SIDE/320: GAPH=SIDE/200: GOSUB 350: GO
TO 130
480 CLS: BEEP: PRINT "ERROR! File not found. "
490 CLOSE: GOTO 20
500 OPEN "1",#1,"FRAC.DAT"
510 OPEN "2",#2,"COPY"
520 IF EOF(1) GOTO 650
530 INPUT#1,AA$
540 INPUT#1,AC
550 INPUT#1,BC
560 INPUT#1,S
570 INPUT#1,MM
580 IF A$=AA$ GOTO 520
590 WRITE#2,AA$
600 WRITE#2,AC
610 WRITE#2,BC
620 WRITE#2,S
630 WRITE#2,MM
640 GOTO 520
650 WRITE#2,A$
660 WRITE#2,ACORNER
670 WRITE#2,BCORNER
680 WRITE#2,SIDE
690 WRITE#2,M
700 CLOSE
710 GOSUB 310
720 KILL "FRAC.DAT"
730 NAME "COPY" AS "FRAC.DAT"
740 RETURN
```

Listing A. (See "Saving fractal parameters," left.)

As many unfinished pictures as you want may be so stored and recalled. When you wish to resume, the computer looks up the parameters in this file and automatically inserts them without the user having to resort to the medieval practice of jotting things down by hand.

This modification leaves the original program almost untouched. By entering anything but a "Y" or "y" at the "Resume old picture?" question, you will be presented with the old routines. If you have any partial pictures already in progress, you can add them to the FRAC.DAT file by calling them up in the old manner (enter "N" at the "Resume?" question) and then entering them by hand. When you then exit via the "Z" command, the current partial picture will be added to the FRAC.DAT file.

William G. Nabor
Mission Viejo, CA

For winners of the Fractal Display Contest (announced in *Sextant* #23), see "Winning Fractals," on page 6 of this issue.

FBE is one of the best

Reference is made to *Sextant* Issue #23, July-August 1986, and the superb article entitled "FBE Research: A Former Boeing Engineer," written by Peter Ruber.

I don't usually write letters to editors, but this is an exception. I just want to give

Listing B. (See "Saving fractal parameters," page 3.)

Z-BASIC VERSION FRACTALS by E. BYRNES. PATCH by B. NABOR
This version has been renumbered. Lines 50, 60, 70, 80, and 90 are new and to be inserted directly after line 40 in the original program. Line 170 (old line 120) has had its GOSUB renumbered. Same for line 200 (old line 150). Lines 460 on are new and to be inserted following the old line 370.

```

10 CLS: PRINT TAB(20) "Fractal Image Generator": PRINT
20 ON ERROR GOTO 450: FILES"??????????.RED": FILES"?????????.GRN": FILES"?????????.LU"
30 PRINT "The picture file(s) above are presently on the current drive."
40 INPUT "Name of this fractal (don't use extension!)";A$
50 PRINT: PRINT "Resume old picture? <Y/N> "
60 Q$=INPUT$1)
70 IF Q$="Y" OR Q$="y" GOTO 460
80 PRINT: PRINT
90 PRINT: PRINT "The recommended lambda values are 0 to -2.0 & 0 to -1.25."
100 INPUT "Lambda and side";ACORNER,BCORNER,SIDE: GAPX=SIDE/640: GAPHY=SIDE/225
110 INPUT "Start at row (row #1 for new image) ";M1
120 PRINT "LOAD partial Fractal image? Y/N?"; ZZ$=INPUT$(1)
130 IF ZZ$="Y" OR ZZ$="y" THEN CLS:GOSUB 410 ELSE CLS:GOTO 140
140 FOR M=M1 TO 224: FOR N=1 TO 639
150 AC=(N*GAPX)+ACORNER: BC=(M*GAPHY)+BCORNER: GOSUB 210
160 AZ=0: BZ=0
170 ZZ$=INKEY$: IF ZZ$="Z" THEN GOSUB 600: END
180 IF SIZEQ>4 THEN PSET(N,M),V
190 COUNT=0: SIZEQ=0
200 NEXT: NEXT: LOCATE 1,1,1: GOSUB 600: END
210 AX=AZ^2-BZ^2+AC: BX=2*AZ*BZ+BC: AZ=AX: BZ=BX: SIZEQ=AX^2+BZ^2
220 IF SIZEQ>4 THEN GOSUB 240: RETURN
230 COUNT=COUNT+1: IF COUNT<100 THEN 210 ELSE RETURN
240 IF COUNT>50 THEN V=5
250 IF COUNT>25 AND COUNT<=50 THEN V=7
260 IF COUNT>15 AND COUNT<=25 THEN V=4
270 IF COUNT>8 AND COUNT<=15 THEN V=6
280 IF COUNT>7 AND COUNT<=8 THEN V=2
290 IF COUNT>5 AND COUNT<=7 THEN V=1
300 IF COUNT>3 AND COUNT<=5 THEN V=3
310 IF COUNT>2 AND COUNT<=3 THEN V=1
320 IF COUNT>1 AND COUNT<=2 THEN V=3
330 IF COUNT>0 AND COUNT<=1 THEN V=1
340 RETURN
350 DEF SEG=&HE000: BSAVE A$+"_.GRN",0,64000!
360 DEF SEG=&HD000: BSAVE A$+"_.RED",0,64000!

```

For the Z100...

WatchWord®

The ultimate in word processing with speed and power. See subscripts, superscripts, underlining, and boldface directly on the screen. Create your own fonts and special characters. Other features include centering, formatting, automatic horizontal scrolling with long lines, large file capability, split screen, macros, color, and an extensive configuration facility. See reviews in *Remark* (July 1985) and *Sextant* (Jan-Feb 1985, Sep-Oct 1985). Requires 192K RAM.

\$100.00

The Resident Speller™

Spelling checker for use with WatchWord. Checks as you type from inside WatchWord or checks a file. Includes a 50,000 word expandable dictionary. Requires 192K RAM to check a file. Requires 300K RAM to check as you type.

\$100.00

Demo disk for both

\$ 3.00

S & K Technology, Inc. Quality Software for Heath/Zenith Microcomputers

For IBM compatibles including the Z150 and Z200 series...

PC WatchWord® (New)

\$ 99.95

The ultimate in word processing for the sophisticated user. Most of the features of the Z100 version except for screen fonts. Requires 256K RAM.

PC Resident Speller™

\$ 99.95

Spelling checker for ASCII files such as those created with WatchWord, WordStar, WordPerfect, PeachText, and VolksWriter. Includes Strike. Requires 256K RAM.

Strike™

\$ 49.95

Adds as-you-type spelling checking to your word processor. Works with the word processors above and also with DisplayWrite, MultiMate and PFS:Write. Requires 100K RAM in addition to that used by your word processor.

Demo disk for Strike and The PC Resident Speller

\$ 2.00

Demo disk for PC WatchWord

\$ 2.00

Texas residents please add state sales tax.

S & K Technology, Inc., 4610 Spotted Oak Woods, San Antonio, Texas 78249, (512) 492-3384

Circle #230 on Reader Service Card

```

370 DEF SEG=&HC000: BSAVE A$+"BLU",0,64000!
380 PRINT "NAME: ";A$;"": PRINT "LAMBDA: ";ACORNER;" ";BCORNER
390 PRINT "SIDE: ";SIDE: PRINT "ROW: ";M: PRINT "(Write": PRINT "this": PRINT "down"
n!""
400 RETURN
410 DEF SEG=&HE000: BLOAD A$+"GRN",0
420 DEF SEG=&HD000: BLOAD A$+"RED",0
430 DEF SEG=&HC000: BLOAD A$+"BLU",0
440 RETURN
450 IF ERR=53 THEN PRINT "No Picture files found!": RESUME 30
460 OPEN "I",#1,"FRAC.DAT"
470 IF EOF(1) GOTO 580
480 INPUT#1,AA$
490 INPUT#1,ACORNER
500 INPUT#1,BCORNER
510 INPUT#1,SIDE
520 INPUT#1,MM
530 IF A$<>AA$ GOTO 470
540 M1=MM
550 CLS: GAPX=SIDE/640: GAPY=SIDE/225
560 GOSUB 410
570 CLOSE: GOTO 140
580 CLS: BEEP: CLOSE: PRINT "ERROR! File not found!!"
590 GOTO 20
600 OPEN "I",#1,"FRAC.DAT"
610 OPEN "I",#2,"COPY"
620 IF EOF(1) GOTO 750
630 INPUT#1,AA$
640 INPUT#1,AC
650 INPUT#1,BC
660 INPUT#1,S
670 INPUT#1,MM
680 IF AA$=A$ GOTO 620
690 WRITE#2,AA$
700 WRITE#2,AC
710 WRITE#2,BC
720 WRITE#2,S
730 WRITE#2,MM
740 GOTO 620
750 WRITE#2,A$
760 WRITE#2,ACORNER
770 WRITE#2,BCORNER
780 WRITE#2,SIDE
790 WRITE#2,M
800 CLOSE
810 GOSUB 350
820 KILL "FRAC.DAT"
830 NAME "COPY" AS "FRAC.DAT"
840 RETURN

```

credit where it is due. In my estimation, Dave Brockman and the company he founded is certainly one of the *very best* Heathkit-associated vendors. He is a person with high integrity, and he shows enthusiasm and an interest for his customers. I can attest to the fact that he willingly helped me considerably with my software problems when I first added Spooldisk 89 to my system, and from that time it has been on the go without any problems.

Dave Brockman has a special gift for designing excellent products, and demonstrates compassion to help solve the problems of his customers. If he ever slowed down enough to produce another accessory for this H89, you can bet that I would purchase it.

Dan Jerome
Burnsville, MN

Another vote for "The Eight-Bit World"

Walter Janowski's "The Eight-Bit World" column in *Sextant* is a wonderful and much needed source of information for those of us who continue to look for ways to revitalize our H89s, and thus save them from the electronic scrap-heap. I sincerely hope that you will continue to be sensitive to the needs of the eight-bit user/hobbyist in the future.

James Lubbe
Madison, WI

ZRAM-205: RAM Multiplier

The ZRAM-205 package is used to modify a Z-205 memory card so that thirty-six 256K RAM chips (not included) can be installed in place of the existing 64K RAM chips giving 1024K of bank-switched memory in the form of four banks of 256K each. Bank 0 (256K) is selected automatically on power-up and must be used as part of system memory. Included MS-DOS software (with source code) makes the remaining three banks into a 768K RAM disk. The modified Z-205 still uses only one S-100 bus slot and will operate at 8MHz if 150 nanosecond RAM chips are used.

The ZRAM-205 package includes a small piggy-back circuit board, a packet of wire and sleeving, a diskette, and a user's guide. Installation takes about 2 hours and requires some soldering to connect the RAM chip socket pin-1's together and to the small circuit board. Our ZP173 PAL is required if used with new motherboard 768K Z-100™.

\$49 Each (+ \$8 for ZP173 PAL)

ZMF100a: 768K On Old Motherboard

The ZMF100a™ modification package lets you plug industry standard 256K RAM chips into your "old" motherboard Z-100™ computer. Up to 768K (27 RAM chips) without using an S-100 bus card slot! Maintains fully compatible memory maps so that all software will run. Usable only on part number 85-2653-1 motherboard (assembly part number 181-4917 or less)!

Installation of the ZMF100a™ package is a simple, plug-in job requiring **no soldering** or trace cutting and only partial disassembly of the computer. The memory chip pin-ones are connected together by a plug-in bus bar. There are no permanent changes to the motherboard.

The ZMF100a™ is a professional quality circuit board supplied with complete instructions and all required materials, except memory chips. It is compatible with the popular PC emulators and with 8MHz speed modules.

\$65 Each

FBE SmartWatch: Calendar/Clock

The Dallas Semiconductor DS1216E SmartWatch is a chip-sized module that plugs-in under the computer's monitor ROM. It contains a full-functioned calendar/clock and works in the Z-100 series (spacer kit required), Z-138/148, Z-150/160 series, and Z-158 computers. Easy installation. Our package includes the SmartWatch, MS-DOS date/time setting software on diskette (with source code) and documentation.

\$44.95 Each (+ \$2.00 for Z-100 Spacer Kit)

FBE

FBE Research Company, Inc.
P. O. Box 68234, Seattle, WA 98168
(206) 246-9815

Order Direct from FBE or
See Your Heath/Zenith Dealer.
UPS/APO/FPO Shipping Included.
VISA or MasterCard Accepted.

Trademarks: ZMF100a/FBE Research Co. Inc.,
Z-100/Zenith Data Systems, MS/Microsoft Corp.,
Gemini/Gemini Technology Inc., EasyPC/UCI Corp.,
PAL/Monolithic Memories Inc.

Winning Fractals

The Fractal Image Contest announced in our July-August 1986 issue resulted in a variety of fascinating displays. Here are the winning images, along with some imaginative modifications and variations on the original program.

Edward A. Byrnes

Imagine a small room with three Heath computers, each on a long but cluttered desk. The monitors of two of the computers are displaying colorful fractal images in different stages of completion. The third monitor holds the words you now read.

Near each computer are stacks of disk mailers, disks, letters, and program listings. I'm sipping coffee between writing sentences and listening to the fan on the Z100. (I never did replace it.) As always, I'm wondering where to begin.

The "BASIC Fractals" article that appeared in *Sextant* #23 (July-August 1986) invited you all to enter the "Fractal Image Contest." I was flooded with programs on disk, hard-copy listings, and letters describing enhancements to the programs that were listed with the fractals article.

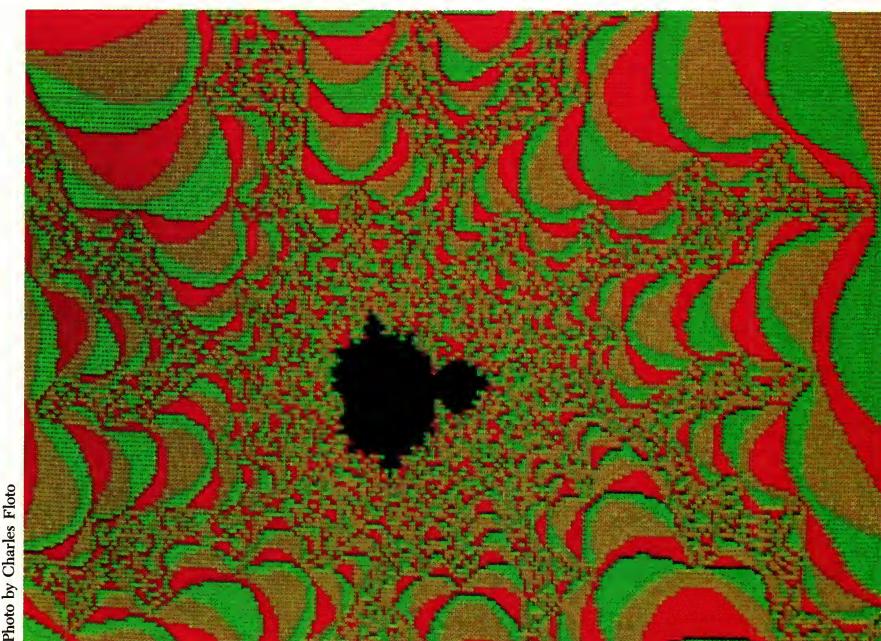
Although the contest announcement mentioned only BASIC, I received many other excellent programs written in assembly language, FORTRAN, Turbo Pascal, Microsoft Lisp, and even Lattice C. All the entries were first-rate, and it was great fun to be able to view the various "explored" areas of the complex plane.

Two winners

Because of the color and resolution disparity between the Z100- and Z150-series computers, winners were chosen for both machines. The programs and their images were judged on beauty, ease of use, and speed.

The Fractal Image Contest winner for the Z100 is Ted W. Miller of Buena Park, California. The winner for the Z150-series computers is Kenneth D. Wright of Kirtland Community College in Grayling, Michigan. Both Ken and Ted sent in a number of images, and they are all interesting.

Edward A. Byrnes is a former philosophy student and Marine. He now divides his time between his software company, Intuitive Logic, in Rochester, Michigan, and working at the local Heath/Zenith store.



This image was generated by Ken Wright's winning fractal program, MandelZoom, for the Z150. (See Listing 2.) MandelZoom allows you to change the color and resolution of an image in successive runs.

One on the Z100

Ted Miller's program (Listing 1) is a modification of my Z100 program that appeared with the "BASIC Fractals" article.

One of the best features of Ted's program is that it mixes the eight colors of the Z100 to produce many other shades (as can be seen in his image on page 7). Horizontal resolution is reduced by half, but the "loss" is more than made up for in each colorful image.

Also, Ted's program stores the picture in a data file—unlike my programs, which store the pictures with BSAVES. When compiled, Ted's program works just as well as mine (and much faster).

And one on the Z150

Ken Wright's program (Listing 2) was inspired by the same *Scientific American* article that inspired my programs. (See A. K. Dewdney's "Computer Recreations" column in *Scientific American*, August

1985.) Ken's program saves images as data and also as BSAVED pictures. In successive runs of the program, you can remanipulate the data to change color and resolution.

Ken's program will create a trouble-free image for redisplay, and you can manipulate the existing data for any image created. If you choose to manipulate the data, a number of parameter questions need to be answered. Accompanying Listing 2 are some brief comments on those questions. (See "Using Ken Wright's Program.")

Ken has also written programs that produce "fractal movies." Small successive changes to the input coordinates are made, for up to 20 images. The images are then loaded into a memory disk; at memory-disk speed, the "movie" program quickly places each image, in sequence, on the screen.

The effect is mesmerizing; the colorful points can actually be seen "fleeting"

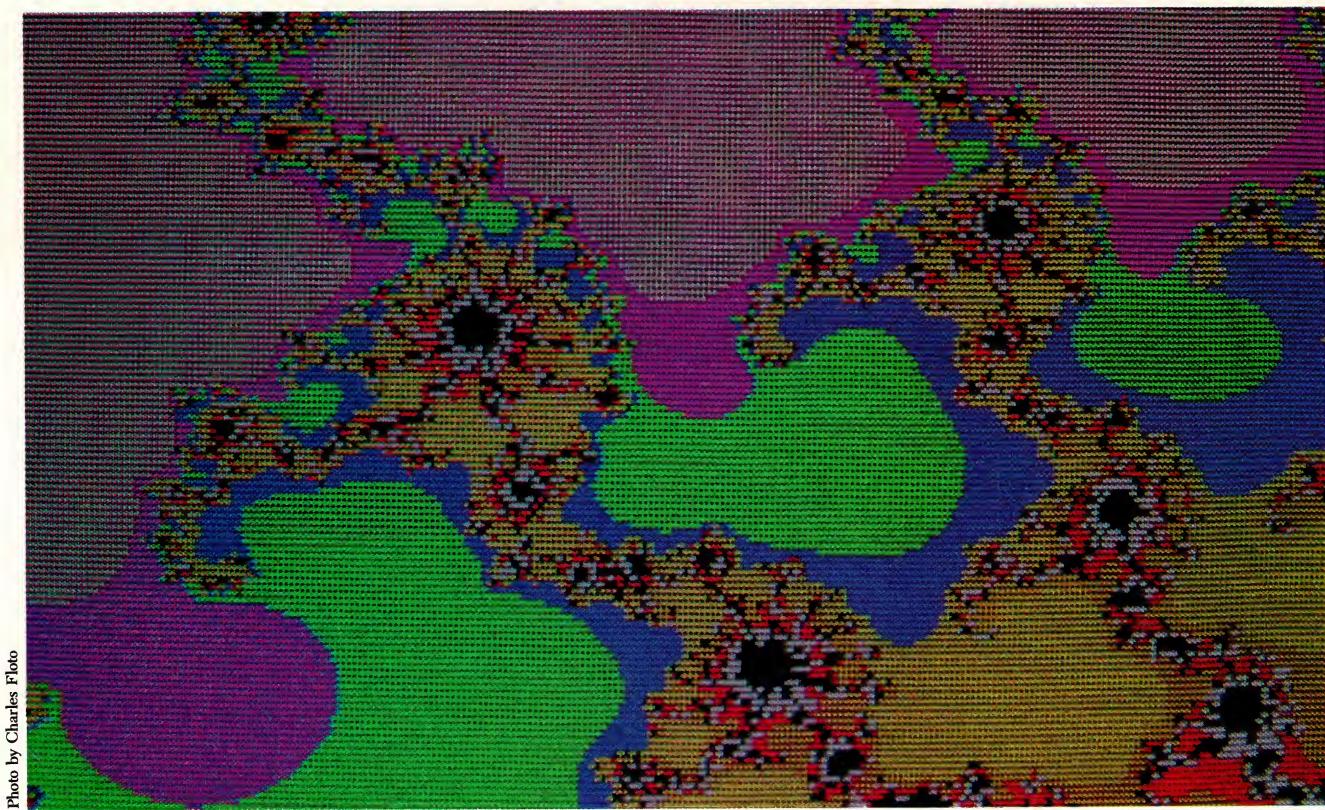


Photo by Charles Floto

Ted Miller of Buena Park, California, was a winner for his images that can be created on the screen of the Z100. Ted's program (in Listing 1) mixes the eight colors of the '100 to

produce many other shades. This program is compatible with the BASIC compiler; in compiled form, it runs much faster.

the Mandelbrot Set. My sons think the movies are awesome, and I think they are most amazing.

Other programs, other enhancements

Jim Reeb of Parker, Colorado, sent me an excellent program that allows more control of parameters and has more nice extras than any of the other programs submitted. It offers parameter control, saving of images as data files, and a feature to track how much time a fractal is taking. Another feature is a cursor that can be manipulated from the keypad. This cursor can be moved around in an existing image to automatically find the coordinates of an area to be magnified.

George Holt of Nashua, New Hampshire, sent some modifications that make the "BASIC Fractals" programs run 20 to 30 times faster. (These modifications appeared in the "Letters" column in *Sextant* #25, November-December 1986.) Among other things, my original program forced a ten-line comparison to determine the correct color value (v); Holt's modification allows program execution to go on as soon as the correct value of v is found.

Stanley Schwartz of Providence, Rhode Island, passed on an idea he used in his Lisp implementation of a fractal-generation program. His program first searches square areas to see whether they are in the Mandelbrot Set; it does so by plotting perimeters. If the entire perime-

Using Ken Wright's Program

Note: To run Ken's program, you must reset the default record size to 640. When you invoke BASIC, use the form:

BASIC COLOR321/S:640

If you wish to manipulate the data for an already created image, the program gives you the following choices:

Quantity of breakpoint: This divides the quantity of iterations into groups for different color arrangements. Good values are 1 to 3.

Upper limit of breakpoint: This allows you to set the maximum number of iterations per pixel within the breakpoints. Good values are 100 to 1,000.

Iteration divisor of breakpoint: The "next" color will be chosen after the number of iterations per pixel has increased n units. Use a value from

1 to 100.

MOD divisor of breakpoint: If you wish to alternately apply all three colors, enter a 3; for two colors, a 2; etc. Good values are 1 to 3.

Color offset (0 to 3) of breakpoint: Sets the way colors are grouped; that is, what color follows the next. Start with 1, then experiment.

Start color at iteration number: Start with 0; for maximum iterations, use 1,000.

Maximum iterations for image: Number of iterations per pixel, up to 1,000.

If any of this is unclear to you, Ken said he will be happy to answer your questions. Send your questions and a self-addressed, stamped envelope to: Ken Wright, Route 2, Box RT-30, Grayling, MI 49738.

Here are some coordinates to get you started with Ken's program:

S.W. corner/real number =	-1.1	-0.7275	-1.94375	-0.1787453
S.E. corner/real number =	-1.0	-0.72625	-1.9375	-0.1738594
N.W. corner/imag. number =	0.35	0.258125	0.0003125	1.074206
Iterations per pixel =	300	300	300	300



Need a PAL® For Your Z-150?

MegaRAM-150™ package allows five banks of 256K RAMs to be installed on standard memory card for over a megabyte of total memory. Resulting memory organized as 640K/704K main and 512K software controlled overlay. Simple installation requires no soldering. Includes software Device Driver to use overlay memory as 512K RAM disk. **\$49.95 each ppd.**

ZP640 PLUS replacement for U455 PAL® allows two banks of 256K RAMs and two or three banks of 64K RAMs to be installed on standard memory board for 640K or 704K total memory [size selected by J401 jumper]. **\$24.95 each ppd.**

COM3 package changes COM2 serial port to new port address (3E8H) and thus allows total of three serial ports. COM2 assumed to be on internal modem card or similar device. Printer driver software included. **\$39.95 ppd.**



FBE Research Company, Inc.

P.O. Box 68234
Seattle, WA 98168
(206) 246-9815

PAL® is a trademark of Monolithic Memories Inc.
VISA and MasterCard Accepted.

Circle #102 on Reader Service Card

Lifetime Guaranteed Bulk Diskettes with Tyvek Envelopes and Labels

Bulk 3M DSDD Disks	Soft Sector	
DSDD 25-\$17.50	50-\$34	
100-64		
1-2 Boxes	3+	
Nashua DSDD \$ 8.95 \$ 8.25		
10 Sector Available on Nashua		
BASF DSDD	9.95	9.50
Xidex DSDD	7.75	7.50
Dysan DSDD	19.95	18.95
Maxell DSDD	12.95	11.95
Dysan Hi Dens.	33.95	32.95
Brown DS 96tpi	23.00	21.50
Brown HI Dens.	24.95	23.95
Maxell HIDens.	24.95	24.25
Nashua HIDens.	23.95	21.40
Paper #20 2500 sheets	\$ 31.50	
9.5"x 11" micro-perf. Delivered		
Perfect data 8" head cleaner		
2 disk wet system	\$ 16.95	
Gemini 5" head cleaner single		
disk wet system	\$ 5.95	
DISK STORAGE		
Ringking 8" w/lock	\$ 28.95	
FlipnFile 8" holds 50	\$ 26.95	
Media mate 5" holds 50	\$ 9.95	
DX-100 5" holds 100	lock \$12.95	
DX-100B 50 ea side lock	\$13.95	
DF-120 holds 120 nolock	\$11.95	
DD-50L holds 50 w lock	\$ 10.95	
Terms: Check or money order! Add		
\$ 1.50 S&H, 48 STATES ONLY!!!		
OTHERS CALL OR WRITE !!!!!!!		
NYS RESIDENTS add sales tax.		

MATCOMM

P.O. Box 604 Bronx, NY 10469
(212) 652-2053 after 7 p.m. best time

Circle #192 on Reader Service Card

```

10 CLS: PRINT "Ed Byrnes Fractal Image Maker"
12 PRINT "Enhancements by Ted Miller... Z-100 version."
20 DEFINT I: DIM IMAGE(122),ICLR(24,2): J$=STRING$(246,32)
22 DATA 2,1,4,4,0,5,6,4,5,8,7,5,10,0,2
23 DATA 12,1,2,14,3,2,16,0,1,18,1,1,20,1,3
24 DATA 23,0,3,26,3,3,30,6,7,35,5,6
25 DATA 40,4,7,45,5,5,50,2,2,55,1,1
26 DATA 65,6,6,75,4,6,85,4,4,99,7,7
28 FOR I=0 TO 21: FOR J=0 TO 2: READ ICLR(I,J): NEXT: NEXT
30 ON ERROR GOTO 410:FILE$"??????????.PAR"
40 PRINT "The picture files above are presently on the default drive."
50 PRINT "Load picture file? (Y or N) ";:ZZ=ASC(INPUT$(1)) AND &H5F:
PRINT CHR$(ZZ)
60 INPUT "Name of this fractal (do not use . or extension) ";A$
70 IF ZZ=89 THEN OPEN "I",1,A$+".par": INPUT #1,ACORNER,BCORNER,SIDE,M1:
CLOSE: PRINT "LAMBDA";ACORNER;,";BCORNER; SIDE=";SIDE;" START
ROW="";M1;" - OK? (Y or N) ";: ZY=ASC(INPUT$(1)) AND &H5F: PRINT CHR$(ZY)
80 IF ZY<>89 THEN INPUT" LAMBDA & SIDE ";ACORNER,BCORNER,SIDE
90 IF ZY<>89 THEN INPUT" START ROW (1 for New Image) ";M1
100 CLS:BCORNER=BCORNER+SIDE:GAPX=SIDE/320:GAPY=SIDE/225:IF ZZ=89 THEN GOSUB 270
110 FOR M=M1 TO 224: BC=BCORNER-M*GAPY: FOR N=1 TO 319
120 AC=ACORNER+N*GAPX: GOSUB 170
140 I$=INKEY$: IF I$="" THEN 430
150 AZ=0: BX=0: COUNT=0: SIZEQ=0
160 NEXT: NEXT: GOTO 430
170 AX=AZ^2-BX^2+AC: BX=2*AZ*BX+BC: AZ=AX:SIZEQ=AX^2+BX^2
180 IF SIZEQ<0 THEN COUNT=COUNT+1: IF COUNT<100 THEN 170 ELSE RETURN
190 I=0: WHILE COUNT>ICLR(I,0): I=I+1: WEND: PSET (2*N-1,M),ICLR(I,1):
PSET (2*N,M),ICLR(I,2): RETURN
270 ' bload files
280 OPEN "R",1,A$+".PIC",246: FIELD #1,246 AS I$
290 FOR I=0 TO 224: GET #1
300 FOR J=0 TO 122: IMAGE(J)=CVI(MID$(I$,2*J+1,2)): NEXT
310 PUT (0,I),IMAGE: NEXT: CLOSE
320 RETURN
330 ' bsave files
340 OPEN "R",1,A$+".PIC",246: FIELD #1,246 AS I$
350 FOR I=0 TO 224
360 GET (0,I)-(639,I),IMAGE
370 FOR J=0 TO 122: MID$(J$,2*J+1)=MKI$(IMAGE(J)): NEXT
380 LSET I$=J$: PUT #1: NEXT: CLOSE
390 OPEN "O",1,A$+".PAR": WRITE #1,ACORNER,BCORNER,SIDE,M: CLOSE
400 RETURN
410 IF ERR=53 THEN PRINT "No picture files found!": RESUME 60
420 PRINT "Error #";ERR;" at ";ERL: END
430 IF I$<>CHR$(3) THEN GOSUB 330
440 LOCATE 1,1,1: END

```

Listing 1. Ted Miller's program is an enhancement of Ed Byrnes's "BASIC Fractals" program from issue #23. Ted's program mixes the eight colors of the Z100 to produce additional shades. This reduces the horizontal resolution of the image, but it results in a much more colorful image.

Listing 2. Ken Wright's program allows you to change the color and resolution of an existing image. (See the comments on page 7 for a discussion of how you can manipulate the data of an existing image.) Ken's program was inspired by A. K. Dewdney's article in the August 1985 issue of *Scientific American*.

```

10 '
***** MANDELZOOM *****
20 '
30 By: Kenneth D. Wright, Kirtland Community College
40 '
50 ' Program suggested by Kee Dewdney, Scientific American, August 1985.
60 'Numbers in the complex plane, named for Benoit B. Mandelbrot. Mathematical
70 'study of forms with a fractional dimension, called FRACTAL Geometry.
72 'Remember to use this program with the form BASIC COLOR321/S:640
80 '
90 ' Initialize variables
100 '
110 DIM A$(320):T$="":P(1)=2:P(2)=1:P(0)=3
120 '
130 ' Menu selections
140 '
150 SCREEN 2:SCREEN 0
160 CLS:PRINT"Enter #1 To generate a new MANDELBROT image."
170 PRINT T$#2 To read iteration count image from disk."
180 PRINT T$#3 To read screen image from disk."
190 PRINT T$#4 To terminate program.":PRINT
200 PRINT"Enter your selection.....":B$=INPUT$(1):CLS
210 ON VAL(B$)GOTO 310,310,260,220:GOTO 150
220 STOP
230 '
240 ' Read previously stored BSAVED image
250 '
260 LINE INPUT"Enter screen image file name = ";Z$
270 SCREEN 1:DEF SEG=47104!:BLOAD Z$,0:B$=INPUT$(1):GOTO 150

```



```

280 '
290 ' Name and open your file
300 '
310 LINE INPUT"File name = ";Z$:OPEN"R",1,Z$,640
320 FOR X#=0 TO 199:FIELD#1,X# AS DU$,X# AS DM$,2 AS A$(X#+1):NEXT
330 FOR X#=200 TO 319:FIELD#1,200 AS DV$,200 AS DW$,X#-200 AS
DU$,X#-200 AS DM$,2 AS A$(X#+1):NEXT
340 ON VAL(B$)GOSUB 380,510:GOTO 150
350 '
360 ' Generate a new MANDELBROT image
370 '
380 INPUT"S.W. Corner of Real Number.....=";A:C=A:D=A
390 INPUT"S.E. Corner of Real Number.....=";SE:j=ABS((A-SE)/200)
400 INPUT"N.W. Corner of Imaginary Number.....=";B:E=B
410 INPUT"Number of iterations per pixel...=";I:CLS:SCREEN 1
420 FOR Z#=1 TO 200:FOR Y#=1 TO 320:FOR V#=1 TO I
430 F=A*B:A=A*B*B#-1+C:B=F+F+E
440 IF A#*A+B*B>4 THEN PSET(Y#=1,Z#=1),P(V# MOD 3):GOTO 460
450 NEXT V#
460 LSET A$(Y#)=MKI$(V#):C=C+J:A=C:B=E:NEXT Y#:PUT#1,Z#:
C=D:A=C:E=E-J:B=E:NEXT Z#
470 DEF SEG=47104!:BSAVE"A:SCREENIM.AGE",0,16384:B$=INPUT$(1):CLOSE:RETURN
480 '
490 ' Draw MANDELBROT from disk stored information
500 '
510 INPUT"ENTER QUANTITY OF BREAKPOINTS (10 MAX).....=";Q$:
IF Q$>10 THEN 510
520 FOR X#=1 TO Q#
530 PRINT"ENTER UPPER LIMIT OF BREAKPOINT"X#".....=";:INPUT T$(X#)
540 PRINT"ITERATION DIVISOR (1 TO 100) OF BREAKPOINT"X#".....=";:INPUT D$(X#)
550 PRINT"MOD DIVISOR OF BREAKPOINT"X#".....=";:INPUT M$(X#)
560 PRINT"COLOR OFFSET (0 to 3) OF BREAKPOINT"X#".....=";:INPUT V$(X#)
570 NEXT X#:PRINT
580 INPUT"START COLOR AT ITERATION NUMBER...=";P#
590 INPUT"MAX ITERATIONS FOR THIS IMAGE.....=";I#
600 INPUT"SELECT BACKGROUND COLOR (0 to 15)....=";B#
610 INPUT"SELECT PALETTE OF COLORS (0 or 1)....=";C#
620 SCREEN 1:COLOR B#,C#:FOR X#=1 TO 200:GET#1,X#;FOR Y#=1 TO 320:N#=CVI(A$(Y#))
630 IF N#>P# THEN IF N#<I# THEN FOR Z#=1 TO Q#:
IF N#>T$(Z#)THEN NEXT Z# ELSE PSET(Y#=1,X#=1),V$(Z#)+(N#/D$(Z#) MOD M$(Z#))
640 NEXT Y#,X#:GOTO 470
650 '
660 STOP' This version for ZENITH / IBM...Date of latest revision: 09/13/86
670 '
680 'SAVE"A:COLOR321.BAS",A

```

ter of a square is black, then all the points in the square are in the Mandelbrot Set and are black. Since plotting points in the Mandelbrot Set takes the most time, his program plots only those areas for which the perimeter shows color.

George Crawford of Upper Marlboro, Maryland, has written an excellent fractal program in assembly language. The program uses interlace mode (640 x 480 pixels) on the Z100, and generates fast, colorful, high-resolution fractal images. The program is available for \$19.95. The accompanying documentation and source code illustrate resetting the screen-controller chip, reading and writing directly to the video memory, and compressing files. (Write to George Crawford, 12117 Old Colony Drive, Upper Marlboro, MD 20772.)

Another program you can purchase is from Richard A. Tilden of Somerville, Massachusetts. It's called Mandel, and is written in Z-BASIC. Mandel is available as object code for \$25; a package with both object code and source code is available for \$45. (Write to Richard A. Tilden, Software Toolsmith, 10 Thurston Street, Somerville, MA 02145.)

In the "Letters" column in issue #25, Tilden gave some pointers on fine-tuning the original "BASIC Fractals" programs.

Congratulations to Ken Wright and Ted Miller. My thanks to everyone who took the time to enter the contest, and I hope you've all had a good time generating fractal images. △

Disks \$9

Hard or Soft Sector, Double Sided, Double Density; Hub Rings, Write tabs, Rainbow labels.

HARD SECTOR Single sided, Double Density. \$ 8.00 for ten.

QDC

Quick Disk Copy utility for Z-150 and PC's. Reads your source disk into memory once, then formats, writes, and verifies destination disks. Allows you to make multiple copies of the same source without re-reading the disk. \$ 25.00 postpaid.

Z-UTIL

For Z-100. Includes SCREEN, dump for most popular dot-matrix printers, RUNPC, PC emulator program, and TP, Computer- typewriter program for memos, etc. \$ 35.00 ppd.

User Programmable Character drivers for H/Z-8/89/90. For HDOS or Heath CPM (specify) with Epson, MPI, C.Itoh/NEC 8023, Microline printer (specify). Prints H/Z graphics, and inverse, on your printer. Accepts input from Software Toolworks Ed-A-Sketch. \$ 25.00 ppd.

Ribbons

Ribbon Refills w/ Gloves

#7201 Epson MX/FX/RX,MPI	\$ 3.00
#72011 As above, RED	\$ 3.50
#72012 As above, GREEN	\$ 3.50
#7202 NEC-8023,CItoh 8510	\$ 2.50

Cartridges

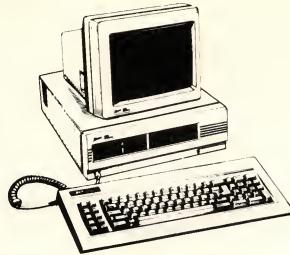
#7203 Epson MX/FX/RX-80	\$ 6.00
#7206 Epson MX/FX/RX-100	\$ 8.50
#7204 NEC-8023,CItoh 8510	\$ 5.00
#7207 Diablo Multistrike	\$ 3.75

Ribbon Spools

#7205 Star SG/Gemini/Oki	\$ 2.50
--------------------------	---------

Please add \$2.00 to your ribbon or disk order for shipping and handling.

LINDLEY SYSTEMS 21 Hancock St.
Bedford, MA 01730 (617) 275-6821



THANKS A MILLION

Thanks a Million to the hundreds of you who participated in our now famous **One Million Dollar Zenith Software Sale**. Of the 49 values in the original offering, only 17 remain. They are listed below at "give away" prices. Many new items are in this ad. We especially want to point out the new items from the UCI Corporation and the SmartWatch from FBE Research. Please **MAIL** or **PHONE** your order today and expect prompt service. **MASTERCARD** and **VISA** gladly accepted with no additional charge.

★★★ ZENITH SOFTWARE FOR THE ★★★ Z-100 SERIES COMPUTERS

Zenith packages with software, manuals and registration cards for the original Z-100 computer series (not for the IBM compatibles).

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-463-1	Z-Basic (16 bit) interpreter	\$175.00	\$24.00
MS-463-7	Multiplan	\$195.00	\$24.00
CB-463-11	Z-Chart	\$150.00	\$15.00
CD-463-2	Condor File Manager	\$299.00	\$24.00
PK-100-4	All 4 listed above	\$819.00	\$62.00
MS-253-1	Microsoft BASIC-80 (8-bit)	\$175.00	\$24.00
OS-53-2	CP/M-85 (8 bit)	\$150.00	\$24.00
OS-63-4	Z-DOS	\$150.00	\$35.00
CB-463-9	PECON Peachtree to Condor	\$99.00	\$15.00
RS-463-1	Peachtree General Ledger	\$399.00	\$38.00
RS-463-5	Peachtree Inventory Management	\$499.00	\$38.00
RS-463-75	PeachText 5000	\$395.00	\$77.00
SC-463-1	SuperCalc	\$195.00	\$38.00
WI-463-1	Remote Batch Terminal Emulator	\$899.00	\$28.00

★★★ ZENITH SOFTWARE FOR THE ★★★ Z-150/160 SERIES COMPUTERS

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
RS-463-75	PeachText 5000	\$395.00	\$77.00
BP-5063-71	BPI Series Self-Training	\$69.00	\$25.00
BP-5063-8	BPI Personal Accounting	\$195.00	\$55.00

★★★ ZENITH SOFTWARE FOR THE ★★★ H/Z-89 SERIES COMPUTERS

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
HES-837-1	Microstat (96 TPI)	\$249.00	\$40.00
HCP-837-1	Pearl #3 Program Generator (96 TPI, Requires C-BASIC)	\$450.00	\$40.00

★★★ ZENITH MONITORS ★★★

ZVM-1360 Long Persistence 13" High Res	\$586.00
ZVM-1330 Color 13" High Res., RGB only	\$437.00
ZVM-1220 Amber Non-Glare Flat Screen	\$99.00
ZVM-1230 Green Non-Glare Flat Screen	\$99.00
ZVM-1240 Amber Flat Screen TTL input	\$158.00
ZVM-1200 Tilt Swivel Base	\$17.00

★★★ ZENITH IBM COMPATIBLE COMPUTERS ★★★

ZF-158-41	One 360K floppy disk drive, 256K RAM, color ready. Retail \$1999.00, Payload	\$1394.00
ZF-158-42	As above with two floppy disk drives and 256K RAM. Retail \$2199.00, Payload	\$1519.00
ZW-158-42	As above but with 20 MEG Winchester Drive and one floppy disk drive. Retail \$2799.00, Payload	\$2040.00
ZFL-171-42 PORTABLE	with Backlighted LCD screen, 256K RAM, two 1/3 height 360K disk drives, ROM based software, Approx. 14.3 lbs. Retail \$2399.00, Payload	\$1663.00
ZF-181-42 LAPTOP PORTABLE	with the amazing Supertwist LCD backlit screen. 640K RAM, two 720K 3.5" disk drives, clock, P&S Ports and more. Expected Retail \$2,399.00. Please call for Payload price and availability.	
ZF-248-81 AT	Compatible with one 1.2 MEG Floppy Disk Drive, 512K RAM, 8 Mhz, 0 Wait States. Retail \$2999.00, Payload \$2191.00. Requires Z-329 or Z-409 Video Card listed below.	
ZW-284-82	AT Compatible with all above plus EGA Color Card, one 20 MEG 85 ms Hard Disk Drive. Retail \$4399.00, Payload \$3062.00	
ZW-248-84	AT Compatible with EGA Color Card and all above plus one 40 MEG 30 ms Hard Disk in place of 20 MEG Drive. Retail \$5699.00 Payload	3941.00

All above come with MS-DOS supplied Zenith. Payload charges only \$5.00 for shipping of prepaid orders. We can enhance the above computers with more memory or larger winchester drives at low costs to you.

★★★ UPGRADE ACCESSORIES FOR Z-150/160 ★★★ SERIES COMPUTERS

SmartWatch from FBE Research. Installs in ROM Socket on CPU Board in Zenith computer series Z-100/150/158/160. This tiny jewel of a product contains a ten year battery and keeps your computer informed of both time and date at each boot-up. Complete instructions and software included

MEMORY KIT #150-256-18. Includes a ZPAL chip which allows use of 256K RAM chips included (18 pieces 256K 150 ns RAM chips). Kit increases 128K memory to 640K or 256K memory to 704K. All chips plug into your existing Zenith Memory Board. Unbelievable but true

Winchester Hard Disk Drive Internal Set-up. Includes Winchester drive, controller/interface card, cables and all hardware. With 20MEG (formatted) drive \$478.00. May be installed in Z-148 using an Expansion Card sold below.

PTZ-148 Expansion Card for Z-148. Includes 2 expansion slots plus a clock/calendar. \$129.00

Winchester Hard Disk Drive External Set-up. Includes Winchester drive, controller/interface card, power supply and case with fan. With 20 MEG (formatted) drive

a clock/calendar

SECOND INTERNAL DISK DRIVE with instructions and hardware

EVERCOM INTERNAL MODEM. Fully Hayes compatible 1200/300 baud with powerful BitCom software included

★★★ VIDEO CARDS ★★★

Z-171-3 Z-171 Video Card RGB/Comp.	\$189.00
Z329 High-Resolution Monochrome	\$182.00
Z-409 Color Graphics/Monochrome for Z-200	\$205.00



★★★ UPGRADE ACCESSORIES FOR Z-100 ★★★ SERIES COMPUTERS

COLOR GRAPHICS UPGRADE. All memory chips (16 pieces 150 ns) required to update Z-100 Series computer for color. Installation instructions included. Order Memory Kit #100-64-16 \$21.50

MEMORY UPGRADE. All memory chips (9 pieces 150 ns) required to upgrade from 128K to 192K RAM. Installation instructions included. Order Memory Kit #100-64-9 \$12.50

ZMF100A by FBE Research. A modification package which allows 256K chips to be used on the old-style motherboard to reach 768K. Simple assembly with no soldering or trace cutting. Compatible with Easy PC and Gemini Emulator. \$60.00 alone or \$148.50 with 27 256K RAM chips included.

SmartWatch by FBE Research. If you don't have a clock for your Z-100, get this one. More details under Z-150 upgrade listings \$44.00

GEMINI EMULATOR BOARD. Makes the Z-100 compatible with the IBM PC library of programs \$432.00

UCI EASY PC. IBM PC Emulator. Makes your Z-100 IBM Software Compatible. Full 8 MEG operation, color graphics and audio compatible. Retail \$699.00, Payload \$477.00

UCI EASY 87. Add an 8087 Numeric Coprocessor. \$69.00 for the board without an 8087 Chip. With 5 MEG 8087 \$197.00 or with 8 MEG 8087 installed \$234.00

UCI MEMORY UPGRADE CARD. We recommend this one highly. The board has sockets for up to 2 MEG of RAM. With no RAM installed \$328.00. With RAM installed and fully tested, 512K \$387.00, One MEG \$446.00, Two MEG \$564.00 Add \$35.00 for EasyDrive RAM Drive Software if desired.

UCI RAMSAVER. Maintains power on UCI MEMORY CARD RAM when computer is off. Save your programs in RAM with your computer off. Payload \$177.00

UCI EASY-I/O. S-100 board that provides IBM PC communications port compatibility with your EasyPC. Easy I/O-1, One Serial Port \$91.00. Easy I/O-2, Two Serial Ports, One Game Port, Clock-Calendar \$127.00

UCI EasyWin. Winchester Drive Systems at reasonable prices. Complete Hard Disk Systems for mounting inside your Z-100. Systems complete with Seagate Drives, 21 MEG \$745.00, 31 MEG \$865.00. System without Drive \$277.00

CDR Z-100 SPEED MODULE. Run your Z-100 Computer at 7.5 MHz. Installs easily with no soldering. Externally switchable between Speed and Normal mode. Payload \$48.00

★★★ UPGRADE ACCESSORIES FOR H/Z-89 ★★★ COMPUTERS

Magnolia Microsystems Double Density Controller. Soft-sectored disk controller. Supports up to four each 5.25" and 8" disk drives. Complete with cables, installation instructions and CP/M \$294.00

Magnolia Microsystems Memory Upgrade Card. Installs in your H-89 to increase memory from 48K to 64K. Payload price \$50.00

INTERNAL DUAL DRIVE SETUPS. Includes two half height double sided disk drives and all hardware and connectors required to mount inside your H-89. Steel mounting shield/case included.

MITSUBISHI MF501 Setup, 48 TPI, 6 MS seek, 320K \$279.00

MITSUBISHI M4853 Setup, 96 TPI, 3 MS seek, 640K disks \$284.00

★★★ HALF HEIGHT FLOPPY ★★★ DISK DRIVES

MITSUBISHI M2896	8"	48 TPI DS/DD	1.2 MEG	\$375.00
MITSUBISHI M501	5.25"	48 TPI DS/DD	320K/360K	\$110.00
MITSUBISHI M504	5.25"	96 TPI DS/DD	360K/1.2 MEG	\$177.00
MITSUBISHI M4853	5.25"	96 TPI DS/DD	640K	\$130.00

★★★ SEAGATE HARD DISK DRIVES ★★★

ST-225	20 MEG Winchester Hard Disk	\$393.00
With Western Digital Controller & Cables	\$478.00	
ST-238	30 MEG, Requires RLL type controller	\$424.00
With Adaptec RLL Controller & Cables	\$528.00	
ST-4026	20 MEG High Speed for Z-200	\$588.00
ST-4038	30 MEG High Speed for Z-200	\$669.00
ST-4051	40 MEG High Speed for Z-200	\$859.00
ST-4096	80 MEG High Speed with Software	\$1425.00

★★★ POWER SUPPLIES AND CASES FOR DISK DRIVES ★★★

Rugged steel construction with heavy duty power supplies. Purchase with drives and we will install drives in case.

Single 5.25" unit for Full Height Drive	\$68.00
Dual 5.25" unit, specify Full or Half Height	\$92.00
Dual 8" unit with fan, for Half Height Drives	\$188.00
Face plate for single drive in dual case	\$8.00
For WINCHESTER Drive, with fan	\$191.00

★★★ CHIP SPECIALS ★★★

The finest RAM available and at PAYLOAD prices. Order one to one thousand chips and add only \$2.00 for shipping.

64K Dynamic RAM, 150 ns	\$1.35 each
256K Dynamic RAM, 150 ns	\$3.28 each
256K Dynamic RAM, 120 ns	\$3.87 each

V-20 CHIPS. High Speed NEC V-20-8 8088 replacement. These run at up to 8 MEG and are said to increase CPU speed 10-30%.

Payload \$14.75

8087 MATH COPROCESSOR CHIPS. Speeds and improves numeric processing. 5 MEG 8087-3 \$129.00, 8 MEG 8087-2 \$165.00

★★★ TOSHIBA PRINTERS ★★★

The new generation printers are here. The Toshiba 3-in-1 printers offer high speed draft quality, near letter quality and high-resolution graphics using an advanced 24 pin dot matrix print head.

	RETAIL	PAYOUT
P-321-P Parallel	\$699.00
P-321-PS Parallel and Serial interfaces	\$749.00
P-321 Tractor Feed	\$99.00
P-341-PS Parallel & Serial, Wide Carrage	\$1149.00
P-341 Tractor Feed	\$199.00
		\$166.00

PAYLOAD COMPUTER SERVICES

15718 SYLVAN LAKE, HOUSTON, TEXAS 77062
PHONE (713) 486-0687

Your satisfaction is guaranteed. All hardware carries a 90 day warranty. VISA and MASTERCARD orders accepted. Add \$5.00 to all prepaid orders for handling and shipping, we pay the balance. Actual shipping costs billed for net billing orders to approved accounts. We accept purchase orders from schools, government and approved accounts. Mail or Phone your order for prompt service.



New Low Priced Products for both CP/M & MS/DOS computers!

TWIST & SHOUT!

Sideways & Banner Printing Utility for Dot-Matrix Printers

TWIST & SHOUT! is two great pieces of software packaged together at one low affordable price!

TWIST lets you print out huge spreadsheets (or any text file) **SIDEWAYS** on your dot-matrix printer. It literally *twists* your printer's output 90 degrees — no more cutting and pasting to put together a fragmented printout. Unlimited width printouts!

SHOUT prints giant banners in letters from two to eight inches high using three special type styles. Make banners, posters and signs with ease. Anything you can type on the keyboard can be **SHOUTed** in gigantic letters!

	January	February	March	April	May	June	July	August	September	October	November	December
REVENUES												
Widget Sales (units)	3654	2345	3213	4322	2811	3120	3000	3120	3213	3322	3432	3543
Price each	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28
Widget Sales (\$)	4677.12	3001.60	4115.20	5332.16	3594.88	3920.00	3600.00	3920.00	4115.20	4332.16	4547.20	4764.32
Gadget Sales (units)	4221	3432	3434	4367	2121	2121	2121	2121	2121	2121	2121	2121
Price each	2.34	2.34	2.34	2.34	2.34	2.34	2.34	2.34	2.34	2.34	2.34	2.34
Gadget Sales (\$)	9877.14	8077.68	8033.56	10686.72	4947.20	5200.00	4800.00	5200.00	5400.00	5600.00	6000.00	6400.00
TOTAL REVENUES	14554.26	11079.28	12150.74	16218.94	7564.88	8120.00	7600.00	8120.00	8230.74	8342.16	8457.20	8564.32

For any CP/M or MS/DOS computer, just

\$34.95

Special: one MS/DOS or ZDOS and one CP/M version for only

\$49.95

SHOUT YOUR MESSAGE IN A BANNER!

Solve Computer Incompatibility!

MEDIA MASTER lets your Z-100 computer or IBM compatible (PC, XT or AT) read, write and format up to 140 disk formats. Exchange information and programs with other computers for just \$39.95.

MEDIA MASTER PLUS

all of the file transfer capabilities of **MEDIA MASTER plus** it gives your IBM compatible the ability to actually *run* CP/M software. All for just \$59.95.

ACCELERATE 8/16 includes **MEDIA MASTER**, an upgraded emulator and a special easy-to-install NEC V20 chip which lets your IBM compatible run CP/M software at maximum speed. V20 chip and software is just \$99.95.

Call or write for full compatibility chart

MILESTONE Automatic Business Project Planner

Originally \$295 — Now only \$99.95!

Never blow another deadline! Whether you're opening a branch office, hiring a sales force or launching a new product, nothing boosts the bottom line like being on time and within budget. Yet that kind of careful planning can take up a lot of your valuable time. **Milestone** brings all of the sure-fire planning methods of Critical Path Management together in one easy-to-use package for your personal computer. Here's how it works: You divide your major project into a number of smaller tasks. At **Milestone**'s prompts you enter the manpower, cost and time estimates for each activity and indicate which tasks must be completed before others can start. Then **Milestone** does the rest, working out your entire schedule and budget (with holidays, sick days and even lunch hours taken into account). **Milestone**'s clear and concise Time Schedule Report gives you a week by week chart showing detailed start and completion dates for each task, critical deadlines, manpower requirements and costs. **Milestone** is available for any CP/M, ZDOS, MS/DOS or PC/DOS computer and any printer. Specify computer.

We accept **VISA, MASTERCARD and AMERICAN EXPRESS**

Order by mail or call our 24 hour toll free order line from anywhere in the US or Canada:

800-628-2828 (Extension 918)

Technical questions, orders: 818-716-1655 (9-5 PST)

Add \$4 per order postage/handling. Overseas, add \$12. US funds only. CA residents add 6% tax (LA County 6.5%)

Circle #198 on Reader Service Card

Zenith: Planning for the Future

The success of Zenith Data Systems is due in part to the adaptability of both the company and its computers. Also, it didn't hurt to plan ahead.

Wayne Rash, Jr.

When a computer company rises from being a small provider of do-it-yourself computer kits to being a major provider of computer hardware to the government, you know that they're doing something right.

Luck, you say? Maybe.

When that same company brings home the largest microcomputer contract in history, becomes 1986's single largest vendor of computers to the government, and sells over two hundred thousand computers in a single year, it's more than luck. It's planning.

If Zenith Data Systems, and its sister company, the Heath Company, excel in one aspect of computer marketing, it's anticipation of change in the marketplace. In short, they plan ahead. They assume that major changes will come, and design their products accordingly. A good example of this approach is in the design of the H/Z150 and its successors.

What's it all about?

The H/Z150 series of computers is unique among clones of the IBM Personal Computer in that it's "bus-based." All of its components are mounted on cards that plug into one single board, the expansion bus. This board is not much more than connector plugs with circuit traces connecting them; as a bus, its job is simply to provide interconnection among the boards that plug into it.

When a bus-based computer is modified in some way, all that has to be changed is the appropriate circuit board that plugs into the bus. In this regard, the '150 harks back to Heath Company's first microcomputer, the 8-bit H8.

Wayne Rash, Jr., is a consultant with American Management Systems in Arlington, Virginia. He's also a freelance writer; his articles appear in computer publications such as Byte and PC Week.

However, "motherboard" designs were used for both of Heath/Zenith's intervening machines, the 8-bit H/Z89 series and the dual-processor 8-/16-bit H/Z100 series. In a motherboard-based design, all the major elements of the computer are gathered together on one board. (The '100 is actually a hybrid, of course, since it uses an S-100 bus for expansion slots.)

For a given design, the basic advantage of a motherboard computer is economy: it's cheaper to put something on one

The advantage of a bus-based computer is that design changes can be made more easily.

board than on two or three. But if you wish to change any one element of the design, it will probably require accommodations for placement, etc., of other elements on the board. And that in turn may require changes in the manufacturing and assembly processes.

The advantage of a bus-based computer is that design changes can be made more easily. The element or elements that are to be changed are already isolated from the rest of the computer.

As any particular upgrade to the '150 became desirable, therefore, only a single circuit board was affected. This allowed much faster response to changing market conditions.

Moreover, when greater component density became possible for the '150, the boards were reworked, and the computer was built on one fewer board. The combination of bus-based design and more powerful chips allowed Heath/Zenith to trim the cost of the original '150 and offer

it as the H/Z158.

This is not to say that Zenith has continued to build every computer as a bus design. The improved chips also let them quickly produce the H/Z148; as a motherboard design, it's less expandable than the '150, but less expensive. (For more on the development of these two machines, you might want to look at James C. Kunze's "New Life for the Z150 from Gate Arrays," in *Sextant* #24, September-October 1986.)

And not all change has been a benefit.

For instance, the flexibility provided by the '150's design allowed Zenith to bring out a transportable computer, the Z160, in very little time. Unfortunately, it did not guarantee that the '160 would be a commercial success.

(Like the '150, the '160 is based on the same 8088 central processor that drives the IBM PC. And like the '150, it is an almost perfect clone of the IBM. For a "portable," however, it was quite a bit heavier than the market seemed to want. See my review in *Sextant* #18, September-October 1985, "Weighing the Merits of the Z160.")

Also, some systems have been developed with less flexibility, primarily because of a need to keep costs down. Examples of this include the Z148 and the portable Z138, another IBM-compatible.

Was it really intentional?

Often, when any company puts together a string of successes like the one Zenith has, the question of intent is asked.

In this case, I asked it. The person who agreed to talk to me about Zenith's planning and positioning for the future is Andrew Czernek, director of marketing for Zenith Data Systems. For several years, Andy has been deeply involved in defining, designing, and readying for market all ZDS products. In many cases, he is the one who decides the form and function of each of Zenith's new

machines.

I asked Andy if Zenith's machines were designed for flexibility. "Absolutely," he said. "That's why we haven't moved away from the bus design, even though a motherboard would be much cheaper."

That flexibility carries over into the design of the Z240 series. Based on the 80286 central processor, this is one of the few IBM PC AT clones with a bus-based design. "The Z240 bus was designed to permit greater speed. It's an extended 16-bit bus."

Czernek confirmed that Zenith places great importance on the ability to react quickly to changing market conditions. It was this ability to change quickly that helped Zenith win the recent military contract for 90,000 Z248s.

The U.S. Department of Defense is not the only customer impressed by Zenith's ability to package a computer to meet the customer's needs. According to Czernek, ZDS now has about 40% of the Canadian educational market. ZDS is growing fast in Europe, as well; but there, it exists on its own. Zenith doesn't sell televisions in Europe, so there's no familiarity with the company as there is in the U.S. Instead, the computer division must make it on its own.

Czernek pointed to a recent blitz of press conferences and product introductions taking place in Western Europe. "We're growing fast in that market, too," he said.

Is it always this way?

Even the best of companies have made mistakes, including Zenith Data Systems. One of the casualties in the product-development process was the ZP-150, Zenith's foray into the lap-top market.

(The ZP-150 is still available—through the Heathkit catalogue and the Heath/Zenith Computers and Electron-

ics stores.)

Manufactured by Mitsui in Japan, the ZP-150 has a 16-line LCD screen; with no disk capability of its own, it has a built in modem and software available to support data transfer to and from a regular desktop computer. This machine comes with several software packages built in, including Microsoft Word and Multiplan.

Unfortunately, the ZP-150 wasn't what a lot of people were looking for. "It turned out that people wanted removable media, and they wanted to choose their own software," Czernek explained. "They also wanted a screen with 24 lines that was easy to read."

As a result, Zenith went into a technology trade with the now-defunct Morrow Designs; this gave Zenith the rights to use the design of Morrow's portable computer, the Pivot. And Morrow used Zenith technology to make the Pivot completely IBM-compatible, and to improve its display capabilities.

In concert, Morrow and Zenith developed the Z171, a dual-drive portable with an LCD screen. The screen is easier to read than other LCDs because it has its own light source. Industry bought the '171 by the hundreds, and government bought it by the thousands. The lessons Zenith learned with all these machines are preparing them for a future in portable computers.

Sometimes the company avoided mistakes by the slimmest of margins. Equally important, the company learns from mistakes, and better products result.

"I wanted to put the screen from the Z171 on to the Z181 when we were designing it," Czernek said. "Fortunately, I lost on that one."

The screen of the Z181 lap-top has drawn accolades from nearly everyone who has used it. It offers higher contrast than most other LCDs and is generally easier to read. However, it costs much

more than the screen on the Z171. Czernek wanted to keep costs down and use the older screen. It turns out that the screen is one of the primary reasons that people are buying the Z181.

The CMOS future

"CMOS is spreading throughout the product line. As it does so, portables will be more economical," Marketing Director Czernek said. Complementary metal-oxide semiconductor (CMOS) devices are necessary for portable computers because they require very little power and produce very little heat. "The 80386 is already a CMOS chip, and I think we'll have CMOS megabyte memory chips before long," he added.

Zenith is planning a big future for computers with the new 32-bit 80386 central processor chip. "We've had an 80386 machine on the drawing board for six months," Czernek said in September.

What does that mean for the future of Zenith Data Systems? Higher speed and more density. The 80386 machines will run more than twice as fast as current machines, Czernek noted, and the storage densities of both chips and mass-storage units will continue to increase.

Regarding mass storage, we may see improvements based on the disks and drives we already have, rather than any radical change in technology. "Optical-storage technology may have missed its window of opportunity," Czernek said. "The magnetic-storage people are making tremendous strides through improvements in coating technology," he added.

Czernek and his employer feel that the future holds bright promise for Zenith. "The modular architecture helps us prepare for what's next," Czernek said. "The industry has stabilized a bit, and we've got a modular design. We might have one or two surprises for you." △

**INTRODUCING
FOR THE Z-100**

Now, you can improve your presentations and reports. Use ShowOff, THE hi-res, graphics program for the Z-100. Easy to use, ShowOff will help you draw, paint and add text to create outstanding graphics.

ShowOff minimum requirements: Z-100 • color video memory • 384K RAM • MS-DOS 2.0 or higher

640 x 480

ShowOff

**HIGH RES
GRAPHIC EDITOR**

U.S. Census Divisions

Price/Gold Prices

Year	Price/Gold
1970	\$35
1975	\$161
1980	\$345
1985	\$616

**ShowOff \$79 • demo disk \$3 • ShowOff with Logitech Mouse \$169
also available for digitizing pads**

HOGWARE COMPANY
470 BELLEVUE • ST. LOUIS, MO 63119
(314) 962-7833

ORDER DIRECT • VISA • MASTERCARD • CHECK • Dealer Inquiries Welcome

With ShowOff, your Z-100 will outperform all popular PC graphic systems. ShowOff offers:

- 640 x 480 resolution
- 92 fill colors
- 92 patterns
- 25 text styles
- compatibility with other graphic programs

I NEED SEXTANT!

Start my subscription right away!

Enter my subscription for:

- 12 issues for \$29.91 (\$34.50 to Canada)
- 6 issues for \$14.97 (\$17.25 to Canada;
\$21.00 overseas via surface mail; \$35.00
via air mail)

Name _____

Address _____

Payment terms:

- Payment enclosed (checks must be in
U.S. dollars payable on a U.S. bank)
- Bill me
- Charge my: Visa MasterCard

Card # _____

Expiration date _____

How did you obtain this copy of *Sextant*?

- Subscription Retail Store
- ZDS Dealer Friend
- Heath Store Other

We'll start your subscription with Issue #27 (March-April 1987), unless you request otherwise.

S26-S

Special Fifth Anniversary Offer

*Extend my *Sextant* subscription at the lowest
rate ever offered! Send me 26 more issues for
just \$55.55 (U.S. only).*

Name _____

Address _____

My check is enclosed.

Charge my: Visa MasterCard

Card # _____

Expiration date _____

Bill me.

*See the back cover of this issue of *Sextant* for
information on this special offer!*

S26-FAO

When mailing in an envelope,
please use ZIP code 20003.



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

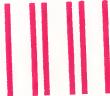
FIRST CLASS PERMIT NO. 13098 WASHINGTON, D.C.

POSTAGE WILL BE PAID BY ADDRESSEE



**BUSS/SEXTANT
716 E Street, S.E.
Washington, DC 20077-0860**

When mailing in an envelope,
please use ZIP code 20003.



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 13098 WASHINGTON, D.C.

POSTAGE WILL BE PAID BY ADDRESSEE



**BUSS/SEXTANT
716 E Street, S.E.
Washington, DC 20077-0860**



YOU NEED **SEXTANT**

Your computer can do a lot for you—if you have the support you need. One source of support thousands of Heath/Zenith users rely upon is *Sextant* magazine—the only independent magazine about your Heath/Zenith computer. Whether you use an H/Z150, H/Z100, H/Z89, or H8, you'll find articles in every issue which apply to your system.

You can explore MS-DOS, Z-DOS, CP/M, and HDOS in *Sextant*. Applications, programs, compatible hardware and software, and the latest developments in the Heath/Zenith community are among the topics covered in every issue of *Sextant*. Regular features include:

"How-to" articles. Designed to help you enhance your system. Read how other users have altered their systems to suit their needs—and how you can too.

Reviews of products from Zenith and independent suppliers. Written to help answer your questions about hardware and software purchases you're considering.

Short program listings. These include utility programs and games, which you can use immediately.

Advertising by independent suppliers. You'll find many products advertised for your Heath/Zenith system that you just won't find anywhere else.

Coverage of community affairs. You'll read about major events, conferences, and personalities in the Heath/Zenith user community, and what effect they could have on you.

Don't miss all the information packed into each issue of *Sextant*!



Start your subscription today!

Call Toll Free:

800/341-1522

(DATATEL 800™, for orders only, Monday-Friday 8 a.m.-9 p.m., and Saturday 9 a.m.-5 p.m., Eastern Time)

Or mail this coupon to:

Sextant, Dept. S26
716 E Street, S.E.
Washington, DC 20003

Please allow 6-8 weeks for delivery of your first issue.

Money-Back Guarantee

Your satisfaction with *Sextant* is completely guaranteed. If at any time you're not completely satisfied, just let us know and your money will be refunded—even for the issues you've already received.

I need *Sextant*!

Send me:

- 12 issues for \$29.91 (\$34.50 to Canada)
- 6 issues for \$14.97 (\$17.25 to Canada; \$21 overseas via surface mail, \$35 via air mail)
- Payment enclosed (checks must be in U.S. dollars payable on a U.S. bank)
- Bill me
- Charge my: Visa MasterCard

Card # _____

Expiration date _____

Name _____

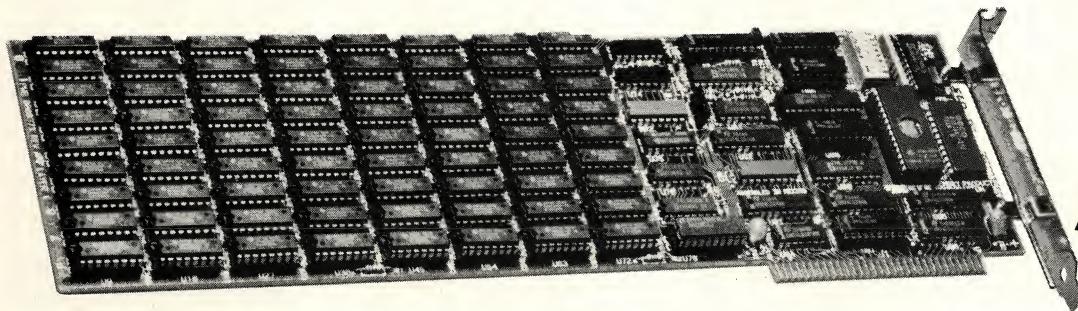
Address _____

Sextant, Dept. S26, 716 E Street, S.E., Washington, DC 20003

Expand your **ZENITH** PC Memory with

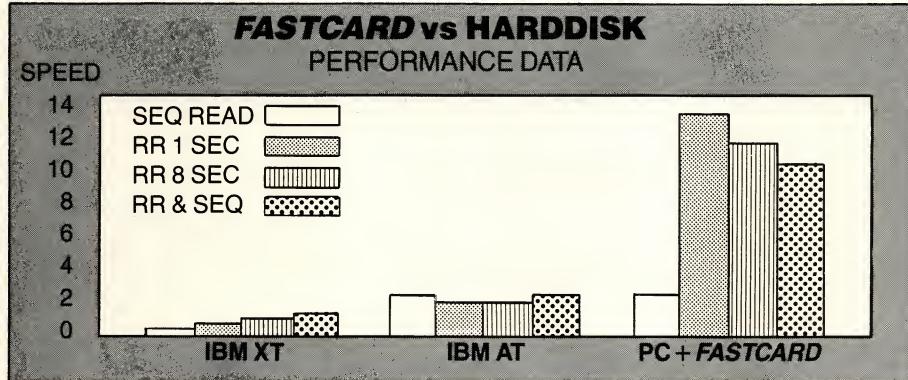
FASTCARD[®]

fully loaded with 2 **MBYTES** for just **\$295***



**10 DAY
FREE
TRIAL!**

**FASTCARD speeds disk I/O
by as much as tenfold,
as shown in the following chart.**



This graph shows benchmark test results of a PC equipped with **FASTCARD** versus a PC-XT and PC-AT. These tests were run by a major independent testing laboratory using their standard disk performance benchmark tests.

Specifically designed for compatibility with the 8 MHz Zenith bus, **FASTCARD** provides both extended and expanded memory plus the following standard features:

- Portable between all Zenith models
- Lotus/Intel/Microsoft compatible
- Up to 2MB with Split Memory Mapping to
 - Fill memory to 640K
 - Provide Expanded memory over 640K
- Unique Disk Caching
- Ram Disks (up to 8MB)
- Custom Password Security
- Print Buffering
- Built-in Diagnostics and Automatic Fault Tolerance
- Factory Installed and Tested DRAMs

*Each **FASTCARD III** comes with 2 MBytes of memory for \$295. **FASTCARD IV**, available with 2 MBytes, includes serial/parallel ports, game port and a clock calendar for \$295.

For additional information, call: **PMI** PERIPHERAL MARKETING, INC. (602) 483-7983
7825 E. EVANS RD., #600, SCOTTSDALE, AZ 85260

Z100 Notebook

William N. Locke

The local computer store

MS-DOS's APPLY utility

The AT utility

ZP/SIM—a CP/M simulator

The Z100 Code Corner

Addressing the Z100 screen

Screen compression

The program

The local computer store

You will be reading this in the winter but, as things would have it, I'm writing it in August.

It was a particularly hot Saturday today, so I thought I would go wander around a computer store and look at the wares. I needed paper and some disks, anyway.

I am getting less and less impressed with these places. The brands of computers are still the same (no Zenith), but the quantity of available software in stock is less and less.

The only compiler in the store was Turbo Pascal. I don't have anything against Turbo, mind you, but one would have expected a little more. C, Lisp, APL, or even FORTRAN would have added variety. There were few games and a limited variety of business and personal-productivity packages.

Could it be that the whole personal-computer hobby will boil back down to a few hackers, like those who started the revolution in the mid-seventies?

I did play with a Panasonic machine; it seemed interesting, but the keyboard left much to be desired. It frustrated me to the point that the salesman came over to find out if I had any idea of what I was doing. "Will this be your first computer?" he asked. "No," I replied, "I have two Zenith Z100s, the ones with the good keyboard."

He grunted. (When will I ever learn?)

The most interesting part of my visit was my experience with their "compatible" store-name brand computer. The machine had the name of their chain and sold for about \$1,600.

I took the system disk out of the Panasonic and tried to boot this new machine. The machine came back with a prompt stating that I had a bad boot disk. I asked the salesman and was told that the machine needed the special store brand of MS-DOS to work. We put the store-brand MS-DOS into the computer and it ran fine.

Now *that* is one poor way to build a "clone." It would be perfectly okay if we were working with an operating system like CP/M; there, you expect that you'll have to buy the operating system in a package specially designed for your brand of computer. But we're not talking about CP/M.

In the IBM-compatible world, too many programs circumvent MS-DOS in order to talk to the IBM hardware directly. The '150 has been designed to look to a program just like an IBM PC; it can run the PC-DOS intended for an IBM. If

you want "compatibility," you need that ability.

There is no excuse for making an "IBM compatible" that checks to see whether some particular company's version of the operating system is the one being booted. When buyers put down their money for such a machine, they ought to be able to select their own software.

Moreover, the store brand of MS-DOS was version 2.1. What if the chain doesn't come out with 3 or 4 or . . . ? What is the customer to do?

If we want to use computers as personal machines, let's not forget that they are expensive toys. The buyer truly needs to beware. The history of people's being burned buying something based on a salesman's hype is simply too long.

I just do not recommend buying a "cheap" compatible. It could be that the best way to buy a computer is to wait for it to be replaced in the marketplace. Computers such as the H89 and the IBM PC jr make superb hobby machines. You can pick up some of these—and even some Z100s—for a song, and there is an enormous amount of inexpensive software available.

I like to encourage computing as a hobby, somewhat like ham radio. The trick is to be able to say how much of your rig you built, or wrote, not how much it cost. Contrary to the opinion of many, one does not need an exceptional IQ to understand programming in assembly language. All that is needed is a good book and a whole lot of patience.

Rather than making an intellectual exercise out of figuring out how some \$500 software works, make the exercise figuring out how the machine works.

If I were to specify the computer store of my dreams, there would be gadgets, robot arms, old and new computers, and plenty of add-ons and software. There'd be a room full of second-hand machines, and a room of buy-it-as-is junk. And don't forget books, magazines, new ideas, and people who like computers. Some computer shows are like that—CHUGCON for one. I truly enjoy that show.

(The Capital Heath Users' Group held CHUGCON 86 on October 25-26, just outside Washington, D.C. For what happened, see "CHUGCON 86: Old Friends and New" by Victoria Saxon, elsewhere in this issue.)

I recognize that my philosophy will not make a computer company rich quickly, but there is a payoff. I am convinced that the success of the Heath (Zenith) computers in the military is due to the fact that

five years ago thousands of military people were building the H89 through the National Technical Schools under the GI Bill.

The people whose views and habits were formed by the hobby environment are in fact the ones providing expertise to current efforts, throughout business, to utilize small computers.

I fear that Zenith may be backing away from the hobby tradition associated with Heath. If they do so, I would ask, "Will the company have anything to distinguish it from IBM?" I would hate to walk into a Heath/Zenith Computers & Electronics store (i.e., a Heathkit store), and find it had turned into a place like that one I walked into this summer.

Time was, I had to drive for half a day to get to a Heath store. It was worth it.

MS-DOS's APPLY utility

The purpose of this and the next Notebook section is to highlight two useful but little-used MS-DOS utilities. The first command to be considered is APPLY; it comes on the MS-DOS system disks. The other, AT, is available as part of the Programmer's Utility Pack. (For a review of the PUP, see "The Programmer's Utility Pack: Useful Tools for the '100 and '150" by William Adney, elsewhere in this issue.)

APPLY will let you perform a specific action multiple times, by supplying a series of different input parameters.

The "action" may be an MS-DOS command, an executable program, or a batch file; it is specified by being expressed in a quoted string. The input parameters may be delivered to the APPLY command from the keyboard, from a file, or through an MS-DOS "pipeline" (discussed below).

The syntax of the APPLY command is basically as follows:

```
APPLY "<command string>" [|S]
APPLY <input_file>
  "<command string>" [|S]
```

In the first case, the command or program in <command string> will take its input either from the standard console input or from a pipeline. In the second, it gets its input from the file referred to as <input_file>.

The quoted string, <command string>, is any allowable command. The string can include a single substitution parameter if desired; this parameter is indicated by the symbol "%". When the APPLY command is executed, the "%" is replaced in turn by each member of a list provided by an input file, the console, or a pipeline.

As an example, if the command string were DEL % and an input of file names were provided, each file would be deleted. (If you include an APPLY line in a batch file, the "%" symbol must be doubled; i.e., "%%" is required.)

The optional \s switch on this command will simply prevent the command lines from being echoed on the screen as they're performed. Without this switch,

each action performed by the command string will be noted on the standard output, usually the terminal screen.

Suppose that you desire to copy all the files on one disk to another, but you want the files on the new disk to be installed in alphabetical order. The following command would perform this task:

```
LS|SORT|APPLY "COPY % A:"
```

LS is a utility from the Programmer's Utility Pack; it produces a list of files from a disk directory. (Here, no drive or file specification is made, so it would be all the files on the current drive.) This list has the file names in the form <FNAME>.<EXT> (such as MYFILE.COM). These are suitable to be input as parameters to other MS-DOS commands.

The list is passed to MS-DOS's SORT utility using the "|" pipeline symbol. SORT puts the list in alphabetical order. Using another pipeline passes the new list to the APPLY command. Finally, APPLY takes each member of the alphabetized list of files and copies it to the disk installed in drive A:.

As a second example, consider the following problem. Let us say that you have a hard-disk partition, G:, with a complicated directory structure. And you want to find a certain file in this structure of directories and place it on a working disk partition that we will call F:. The following command line would perform this function:

```
SEARCH G:MYFILE.EXT | APPLY "COPY % F:"
```

The SEARCH command will search through all the directories of the G: partition until the file MYFILE.EXT is found. Then it will output the file name and the complete required path to that file in a form suitable for input to the copy command. If the file had been three directories down, for instance, the SEARCH command would output something like the following:-

```
|DIRNAME1|DIRNAME2
  |DIRNAME3|MYFILE.EXT
```

This is transmitted to the APPLY command through a pipeline, and the file is copied to the F: drive.

The AT utility

AT has the same basic function as the APPLY command, except that it doesn't support APPLY's switches. A new feature is added, however, that allows the date and time of desired execution to be specified as options.

The AT command would most likely be used for applications related to communications. Say that we want to run a modem program at 8:30 a.m. and we will not be able to be there at that time. The following command would perform that function:

```
AT "MYMODEM"/T:08:30
```

(The time is specified in 24-hour format; so 4:00 p.m., for example, would be 16:00.) To specify the date as well as the time, do the following:

```
AT "MYMODEM"/T:08:30/D:10/25/86
```

This will execute the MYMODEM pro-

gram on October 25th at 8:30 in the morning.

When MYMODEM exits, the AT command will still have control of the computer and will wait for another input. If you have entered MS-DOS's BREAK ON command before the AT command, you can terminate AT with a CTRL-C; otherwise, you have to reset the system.

That reflects the fact that the AT command has a rather unfortunate feature: it steals the computer. While the AT command is waiting for the computer clock to get around to the specified time, you cannot perform any other operations.

I hope to write a modified version of AT, and there may be a future "Z100 Notebook" Code Corner containing it. My goal is to write the new utility so that it will sit in the background and not interfere with computing until the specified date and time. As it is now, the AT command is an almost-great command.

ZP/SIM—a CP/M simulator

When I purchased my first Z100, I had a significant library of CP/M software around the house. I was even one of the few people who managed to latch on to a copy of JRT Pascal years ago before the demand for that product so overloaded the company that it was forced out of business.

As things would have it, I learned to work with Z-DOS, and left those CP/M disks on the shelf. The problem was, however, that most of my games and early development software were out of reach unless I used CP/M and divided my hard disk between two operating systems.

I recently happened on a solution to this dilemma, a program known as ZP/SIM; it was written by Livingston Logic Labs and is currently marketed by The Software Toolworks.

ZP/SIM can be used as an emulator to make the computer act as a CP/M machine, or as a converter to permanently change programs to run under MS-DOS. In either case, you do not need to have a copy of the CP/M operating system.

To run ZP/SIM as an emulator, just type ZPSIM at the MS-DOS prompt; ZP/SIM will then display the prompt, A:}. You are then free to run CP/M programs that have been transferred to your MS-DOS disk via MS-DOS's RDCPM utility. (Before you can run CP/M .COM files this way, you have to rename them with the new extension, .CPM.)

The command processor used by the emulator acts just like ZCPR, the popular public domain enhancement to CP/M. Most of the standard CP/M commands are supported, including DIR, REN, ERA, TYPE, SAVE, LIST, and so on.

It's equally simple to use ZP/SIM to convert CP/M files into executable MS-DOS files. At the MS-DOS prompt, just type ZPSIM <FILENAME>. The file is then changed in such a manner that it will run directly under MS-DOS.

First Capitol Delivers Zenith. Your Way.



Have it your way.

If you're looking for a PC, you couldn't do better than Zenith. And you couldn't buy your Zenith at a better place than First Capitol. We'll put together a system the way YOU want it. And we'll do it for a price that competes with the guy who just sells boxes that fall on his loading dock.

Great Prices. And A Great Machine.

We're committed to Zenith. And to you. the standard Zenith models that come from the factory are a good buy. And we'll be glad to sell you one at a price that will match anyone. But if you want the full potential, we'll fix it up for you. Larger disk drives. More memory. Internal tape back up. Removable winchesters. All at a budget pleasing price.

We make Zenith Scream.

When a new model comes out, we're among the first to get it. Our Zenith fanatics take it apart, prod it and poke it until it screams for mercy. Then we put it back together (usually with some new parts) so that it REALLY screams. With power.

You'll be back.

Check with us. Check elsewhere. We'll put together a custom quotation for you we GUARANTEE you'll find interesting. We think you'll come back to us. There's a reason for the growing list of companies, government agencies, and end users that know us as the source for Zenith. Your Way.



1106 First Capitol Dr.
St. Charles, MO 63301

1-800-TO-BUY-IT (800-862-8948) Orders and quotes

1-314-724-2336 Technical support and order status

The StockSystem Challenge.

We will meet or beat any dealer's price on new stock Zenith systems. And for enhanced systems, no one can touch our combination of value, performance, and custom service.

Before you buy



AUTHORIZED SALES AND SERVICE

Call Us First. And Last.

Circle #206 on Reader Service Card

Function Number	BIOS Purpose
8	Home current drive
9	Select drive
10	Set track
11	Set sector
12	Set DMA address
13	Read sector
14	Write sector
15	Check LST: output status
16	Sector translation
27	Return allocation vector
31	Return DPB address

Table 1. You can get your CP/M programs to run under MS-DOS by using the program ZP/SIM either as a CP/M emulator or as a converter to permanently change your programs to run under CP/M. However, there are some CP/M function calls that MS-DOS will not support. These are listed above.

In either case, much of my old software is back in action.

ZP/SIM is available only for use on the Z100. This is because it uses the 8085 microprocessor, which is not available in other MS-DOS machines. I understand that there is a similar program, Accelerate 8/16, for use on IBM-compatible computers.

Accelerate 8/16 requires that you replace the 8088 microprocessor of the IBM-compatible with a NEC V20—which will emulate the 8080 microprocessor code that CP/M expects. Accelerate 8/16 comes with the NEC V20 chip and some instructions for installing it. I have not tried this program but, like ZP/SIM, it is available from The Software Toolworks.

ZP/SIM does indeed have some limitations. These stem from the fact that the MS-DOS operating system does not support certain of the system services provided by CP/M. Some other functions are only partially supported. (Table 1 gives examples of CP/M function calls that MS-DOS does not support.)

All this means is that operating-system software is liable to not work using the ZP/SIM conversion. Nearly all compilers, editors, business software, and the like will work. Games will work unless they were written expressly for CP/M-85 on the Z100 and rely on the special Z100 clock. Accounting and data base management programs, and programs using H19 terminal graphics will also generally work.

All in all, I would recommend this program to anybody who, like me, wants to bring some old friends back again. I use it with a "try it and see if it works" approach. A few cases (such as with a great CP/M game, YWING.COM) work under the emulation mode while the converted program dies. For \$29.95, the program is well worthwhile. (For more on ZP/SIM, you might want to look at Joe Katz's "Two Ways Your '100 Can Run CP/M Software Under Z-DOS/MS-DOS," in *Sextant* #22, May-June 1986.)

The Z100 Code Corner

The Code Corner this time gives you a program that will read the Z100 screen and place its contents in a disk file. The disk file will hold this screen dump in a compressed form, and thus save a considerable amount of disk space.

To explain the operation of this program, I will first discuss the Z100's screen memory map and how it is accessed. Then I will discuss the compression routines, and finally I will go over the actual program.

Addressing the Z100 screen

The screen of the Z100 is "memory mapped." This means that it is stored in a block of random-access memory (RAM).

Three planes are used (blue, red, and green), each with its own memory block. If a dot (a picture element, or pixel) on the screen is on, it is so because a bit in one or more of these three memory areas is set to one. To read a screen pixel, all that needs to be done is to find the related pixel memory address and determine if the particular bit in the eight-bit word located at that address is turned on.

The mapping of the screen into memory has a special arrangement, and this mapping needs to be explained. The screen consists of 80 columns and 24 (or 25) lines; and each line consists of nine horizontal scans. You can see scans running across the screen as individual dots that make up the characters.

Each character on the screen is located at a specific column and row, and it is eight scan lines tall, including space for any descender. (The remaining scan line represents the space between the rows of characters.) The width of the character is eight bits, which corresponds to exactly one memory byte. Figure 1 shows what each character looks like.

When you are using graphics, the pixels in all nine scan lines can be on. Otherwise, the pattern might be broken at each row boundary.

As you might infer from the description

given above (and as is illustrated in Figure 1), there are some scan addresses that are undisplayed, or "virtual." These can be addressed just as if they were scans 9 and 10, but they will not be displayed.

What would be the tenth scan from the top, marked 9, holds the character font index. This number when added to 32 will yield the ASCII decimal code for the character displayed in this position. The eleventh scan line, marked 10, is the character-attribute byte. (We will not be using the attribute byte here. It is explained in detail by Gary Cramblitt in "Probing the Subconscious Memory of the Z100"; see *Sextant* #24, September-October 1986.)

It should now be easy to understand how to find the addresses of the scan lines in a character. But first a word is needed on the organization of the Z100's memory.

The Z100's microprocessor, an Intel 8088, addresses any given location in memory using two sixteen-bit values: the "segment" provides a base value; and the "offset" is the distance in memory from this base to the address of interest. These values are stored in the segment register and the offset register, respectively.

The actual addressing is done with a twenty-bit value that can be calculated as follows: (1) shift the segment value to the left by four bits; (2) fill the resulting holes with zeroes; (3) add the resulting value to the offset. (See Figure 2.)

Each Z100 color plane has its own segment value, or base address. These are as follows:

Blue segment = C000 (hex)

Red segment = D000 (hex)

Green segment = E000 (hex)

To get to a particular character scan section (one byte), an offset must be computed.

(The offset will be the same for all three color planes. If we add this offset to the segment base address for the green plane, we will be looking at the green section of the character scan line. If we use the blue segment address with the same offset, the blue section of the character scan line will be addressed. For a character to be white, all three planes must have the appropriate bytes adjusted.)

To compute the offset for a particular scan line, the Z100 uses a scheme that is quite easy for the computer to execute—but less easy for us to understand. The offset is sixteen bits wide; the row, scan, and column each occupies a separate group of these bits. They are arranged as shown in Figure 3.

To compute the offset, first adjust the row and scan values: shift the row value left eleven times; shift the scan value to the left seven times.

Then perform a logical OR between the two resulting values, then between that result and the column value. (1 ORed with either 1 or zero produces a 1; a zero is produced only by ORing two zeroes.)

In C, `<<` indicates shifting bits to the left, and `|` indicates performing a logical OR on the bits. So the process would become:

$$\text{offset} = (\text{row} \ll 11) | (\text{scan} \ll 7) | \text{column}$$

With the offset in hand, you can finally read a particular character scan line. In C, you can use the `_peek()` function. With the DeSmet C compiler, you could use:

```
char_scan_byte = _peek( offset,  
    segment )
```

Before you can access the screen directly, the Z100 must be put into a special mode. This is done by sending a code to port D8 (hex). If color graphics is to be used, the code to be sent is 78 (hex). For monochrome, the appropriate code is 08 (hex). In monochrome, all three planes are written whenever any one plane is written. This produces a white pixel even if a bit in only the green plane is accessed. In C (using the DeSmet compiler), this port access is accomplished as follows:

```
For color:  
_outb( 0x78, 0xD8 );  
For black and white:  
_outb( 0x08, 0xD8 );
```

The port-accessing functions vary from compiler to compiler. Consult your specific compiler documentation.

Screen compression

A Z100 screen will occupy a rather large file on the disk. This can be seen by computing the number of bytes that are going to be needed to store it. Each plane will occupy: (25 lines) * (9 scans) * (80 columns) = 18,000 bytes. All three planes will take 54,000 bytes.

This number can be greatly reduced if you take advantage of the fact that many of the adjacent bytes are the same. Rather than having each byte be stored as it is read, bytes can be stored in two-byte groups on the disk. One records the byte value, and the other records the number of times that the byte is repeated in succession. When the groups on disk are later read, each byte is sent to the screen in accordance with its repetition value.

For nearly all graphics screens, this strategy will reduce the screen storage space required to less than 15 kilobytes. However, this compression will fail if all adjacent bytes are different. In that case, the file would be enormous: about twice the 53K required to store the uncompressed screen. A screen full of typed text would suffer from this problem.

The program in Listing 1 gives one solution to this text problem. When dealing with text screens, we don't have to read all nine bytes. Instead, we can simply read the character font index byte (as noted above, 32 less than the character's ASCII value). This way, the compression routine will recognize when two adjacent characters are the same. If GRAPHICS is set to zero in Listing 1, this text mode will be used. The characters will later be

	Ø	x x x x x x x x	(one row space byte
	1	x x x x x x x x	(one character byte
	2	x x x x x x x x	(one character byte
Nine scan	3	x x x x x x x x	(one character byte
lines high	4	x x x x x x x x	(one character byte
	5	x x x x x x x x	(one character byte
	6	x x x x x x x x	(one character byte
	7	x x x x x x x x	(one character byte
	8	x x x x x x x x	(one character byte
Not displayed	9	x x x x x x x x	(character font index
Not displayed	10	x x x x x x x x	(attribute byte

Figure 1. Each line on the screen of the Z100 is made up of nine horizontal scan lines. A character is eight scan lines tall, and the top scan line represents the space between the rows of characters. The two undisplayed scan lines hold information about the character displayed.

Figure 2. The twenty-bit address of the character scan line is calculated as follows: shift the value in the segment register four bits to the left, fill the resulting holes with zeroes, and add that value to the value in the offset register.

offset = $x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14} x_{15} x_{16} x_{17} x_{18}$
| row | scan | column

Figure 3. In the sixteen-bit offset register of a character scan line, the left five bits represent the row value, the middle four bits represent the scan line value, and the last seven bits represent the column value.

Listing 1. If we wish to store a Z100 screen display on disk, simply reproducing all the values in video memory can take up a lot of disk space. This screen-filer demo routine compresses the file by taking advantage of the fact that bit values and character values tend to repeat somewhat. (See text.)

replaced on the screen by adding 32 to the index number and using `putchar()` to put the character on the screen.

The program

The program shown in Listing 1 will read the screen and store it on a disk file. It will then clear the screen and rewrite it with the contents of the file. (Listing 1 generates a pattern of X's to be used as the screen display.)

The functions `fput_scn()` and `fget_scn()` are used to do the screen compression and restoration. Note that the count is stored in a character type variable, so it requires only one byte of file storage. This means that the maximum number that can be used for a byte count is 255. After that, a new value/count pair will be started even if the screen byte has not changed.

The function `service_scn()` does both the screen reading and writing; which one it does at any time depends on the value of the variable `write_flg`, which is set by the functions `write_scn()` and `read_scn()`. If the `write_flg` is zero, `service_scn()` will simply read the entire screen and pass the bytes to the `fput_scn()` function.

Reading the screen and passing the bytes is done by a set of four looping control structures. The outermost is a `do...while...` loop that works through the three color planes. If your machine has just a monochrome video memory, or if you're using a monochrome monitor, you can set the macro `COLOR` to zero; then, only the green plane will be read. The inner three loops work through all the rows, columns, and scans of the screen. In text mode (`GRAPHICS` equal to zero), only scan 9 is read.

If the `write_flg` variable is one, the screen will be written. In graphics mode, this is done in a fashion that allows you to combine several screen pictures into one. The byte at the specified offset is read from the screen; the byte to be placed on the screen is obtained from `fget_scn()`; the two bytes are combined using OR logic; and the new byte is written back to the screen. This allows you to overlay several screen files.

In text mode, the cursor is placed and the character is output using `putchar()`.

Note that in the example, an open file channel is passed as an argument to the `read_scn()` and `write_scn()` functions. Also note that this file channel must be closed once the file is written and before it is read.

Ordering Information

ZP/SIM (for H/Z100 only), \$29.95.
 Accelerate 8/16 (for H/Z150), \$99.95.
 The Software Toolworks
 1 Toolworks Plaza
 13553 Ventura Blvd.
 Sherman Oaks, CA 91423
 Orders, 800/223-8665; in California,
 800/228-8665

```
***** fput_scn() *****
  This function is used to compress the screen data and transmit
  it through the normal C library routine to the disk file. The data is
  stored in groups of two bytes, ( COUNT, CHARACTER STORED).
*****/
```

```
static put_count = START_MARKER;      /* Initialize to start processing */
static fput_scn( byte, f_out )
unsigned byte;
FILE *f_out;
static unsigned put_prev; /*This is used to hold the repeated character */
if( byte!=put_prev||put_count>=MAX_COUNT){/*Not same as old or too many */
  if(put_count != START_MARKER ) /* At start do not put data to file */
    if( fput(put_count,f_out)== EOF||putc(put_prev,f_out )==EOF)
      return(ERROR); /* data pair is out to file, return if error */
  put_count = 0; /* Restart the count at zero */
  put_prev = byte; /* Start counting with the new byte */
}
++put_count; /* Up the count by one. This will give one on first pass */
return(OK);
}

***** fget_scn() *****
  This function is used to decompress the screen information.
*****/
```

```
static get_count = 1; /* The count used for producing the bytes */
static unsigned fget_scn(f_in)
FILE *f_in;
static unsigned get_prev;
if( !(--get_count ) ) /* If get_count drops to zero get new data pair */
  get_count = fgetc(f_in); /* Get count value */
  get_prev = fgetc(f_in); /* Get screen data value */
return(get_prev); /* Return the screen data value */
}

***** service_scn() *****
  This function is the actual screen reading and writing function.
  If the flag, "write_flg", is zero the screen is read. Otherwise the
  screen is written.
*****/
```

```
static unsigned segment,offset,row,column,byte,write_flg;
static service_scn(f_in_out)
FILE *f_in_out;
register srow,scan;
segment = 0xE000; /* Start with the green segment */
do{ /* Do loops through the colors. See "while" below.
  for(row = START_ROW;row<= END_ROW;+row){
    srow = row<<11;
    for( column = START_COL; column <= END_COL; ++column)
      for( scan=START_SCAN; scan<=END_SCAN; ++scan){
        /* offset is: row << 11 | scan <<7 | column */
        offset = srow|scan<<7|column; /* Data offset in segment */
        byte = _peek(offset,segment);/* Read existing screen byte */
        if(write_flg){ /* Writes the disk file to the screen */
          /* Graphics mode recall of recorded data */
          _poke(byte|fget_scn(f_in_out),offset,segment);
        #if GRAPHICS
        #else
          /* TEXT mode recall of recorded data */
          putchar(32 + fget_scn(f_in_out));/* Output char
          if(column == END_COL && row != END_ROW)
            putchar(10);/* Line feed at end of text line */
        #endif
        }
        /* Else place the byte value on the disk */
        else if(fput_scn(byte,f_in_out) == ERROR) return ERROR;
      }
    /* The above lines do screen write and read. Graphics */
    /* write is done by ORing the existing byte with a byte from */
    /* the file. Read is done by passing the byte to put_scn() */
    /* Text mode write is done by adding the font table */
    /* displacement to 32 to get an ASCII value and outputting */
    /* the character at the appropriate row and column position. */
  }
  /* Flush out bytes at end */
  put_count = START_MARKER; /* Set up for another pass */
  get_count = 1;
  }while( COLOR && (segment -= 0x1000) >= 0xC000 );/* If not COLOR QUIT */
return OK; /* Line above steps down through 0xE000, 0xD000, and 0xC000*/
```

```
***** write_scn( f_in ) *****
  These are the intended user functions. f_in and f_out must be file channels
  opened with fopen(). fclose() must be used when the file operation is
  complete.
  write_scn() - This writes a disk file to the screen.
  read_scn() - This reads the screen and places the data in a file.
  Zero is returned if a file i/o error occurs. Otherwise a one is returned.
*****/
```

```
write_scn(f_in ) FILE *f_in; { write_flg = 1; return service_scn(f_in); }
read_scn( f_out ) FILE *f_out;{ write_flg = 0; return service_scn(f_out); }
```



CCS
HAS EVEN
BIGGER
BYTES
FOR YOUR
BUCK!

WE HAVE THE WINCHESTER SOLUTION AT THE RIGHT PRICE FOR ALL YOUR DATA STORAGE NEEDS

For the PC and AT compatibles

GOLD CARDS-Fixed Winchester drives on a full-sized PC card:

21 Megabytes (Lapine Titan, 65ms)	\$599.00
32 Megabytes (Lapine Titan with RLL)	\$799.00

36 Megabyte Fixed Winchester kits (features the CCS-XE board):

CCS-230S (Seagate ST-225, 80ms)	\$699.00
CCS-230L (Lapine Titan, 65ms)	\$749.00

40 Megabyte Winchester for the AT compatible: **\$1299.00**

Removable Winchester storage for the PC compatibles:
 10 Megabytes per cartridge, perfect backup medium:
 CCS-230R (SyQuest SQ312) **\$899.00**

And for you PC owners who already own a Winchester disk, we offer a cheap way to nearly double your disk's capacity:

The Winchester doubler board!
 This board increases a 21 megabyte hard disk to 36 Megabytes! It also features:
 Exceptional error correction capabilities
 Dynamic disk optimization to ensure top performance
 On-board disk caching for maximum speed
 CCS-XE **\$299.00**

For the H/Z-100 series

21 Megabytes of fixed storage:
 CCS-2000S (Seagate ST-225, 80ms) **\$749.00**
 CCS-2000L (Lapine Titan, 65ms) **\$799.00**
 *Price Reduced (Save \$50.00)

Removable Winchester storage:
 10 Megabytes per cartridge, virtually unlimited total storage.
 Perfect backup medium for Winchester system.
 CCS-2000R (SyQuest SQ312) **\$1099.00**

Combo-Get both fixed and Removable storage for your H/Z-100:
 CCS-2000C (Lapine Titan, SyQuest SQ312) **\$1499.00**

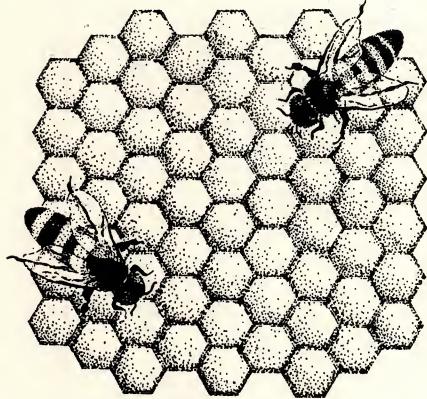
More Bytes From:

CCS

CUSTOM COMPUTER SYSTEMS

P.O. Box 797
 810 Blue Mountain Rd.
 Lyons, Co. 80540
(303) 823-5394

Beyond IBM PC Compatibility:



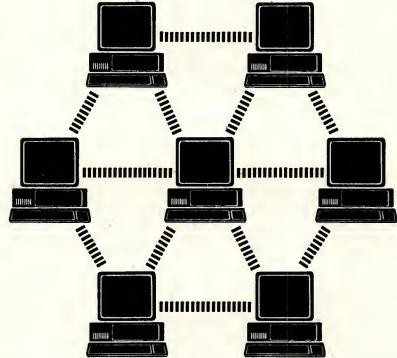
Z-100 users who are ready for PC compatibility, should keep one thing in mind. The future. New, state-of-the-art PC software is creating ever greater performance demands on computer hardware. Large scale, memory intensive programs for networking, graphics, number crunching and communications are quickly becoming the rule, not the exception. But software is only as good as the system that runs it.

EasyPC will run it all.

Even the newest copy-protected best sellers. That's because you actually boot up PC hardware, not software translation routines.

EasyPC runs faster, too.

Faster than an IBM PC, faster than a Z-150, and faster than any other emulation scheme you can buy.



EasyPC will support a fully integrated Local Area Network System (LANS) of Z-100's, PC's and AT's communicating at 115Kbaud.

Networking is now a reality for Z-100 users.

EasyPC will fully support a network of up to sixteen Z-100's, PC's and AT's. EasyPC is the core of an integrated system of components that includes RAM expansion, I/O, math co-processing, hard disk subsystem and more. For serious users, it's the only choice.

Ask to see an EasyPC in action.

Visit your Heath/Zenith Electronics Center or independent UCI dealer and see how easy it is to turn your Z-100 into a PC compatible system. For the name of your nearest dealer, just call 800-UCI-COMPUTER.

EasyPC is a trademark of UCI, Corp. Z-100 and Z-150 are trademarks of Zenith Electronics Corp. IBM and IBM PC are trademarks of International Business Machines.



CORPORATION

948 Cherry Street, Kent, Ohio 44240
(216) 673-5155 / 800-UCI-COMPUTER

Circle #180 on Reader Service Card

The Programmer's Utility Pack: Useful Tools for the '100 and '150

IBM's got nothing that compares to Heath/Zenith's \$225 PUP for MS-DOS Version 3. Here's a description of what the PUP contains.

William M. Adney

If you have Version 3 of the Microsoft Disk Operating System (MS-DOS), perhaps you should consider getting Heath/Zenith's MS-DOS Version 3 Programmer's Utility Pack (PUP). At \$225, this is the most comprehensive programming "kit" that is available in the microcomputer industry today at any price.

I know that is a broad statement, but after you see some of the programs and documentation that are contained in Heath/Zenith's PUP, I think that you will agree. And although the PUP is obviously intended for assembly language programmers, its documentation contains information that is essential to an understanding of how your operating system interfaces to the hardware.

It is also important to recognize that information in the PUP manuals applies to both the H/Z100 series and Zenith's IBM compatibles. However, you will find that the appropriate technical manual for your system is also quite essential if you intend to do any serious programming.

(The IBM compatibles are the H/Z138 through H/Z248 computers, which have also been referred to as the "Z100 PC" series or "Z200 PC" computers. The "Z150 series" includes the Z138 and Z148, which were based on the design of the original Z150.)

In this article, I will take a look at the contents of the PUP in some detail. After all, the list price of \$225 may make it

William M. Adney is a faculty member at the University of Texas at Arlington. He wrote the FlipFast guides to CP/M and MS-DOS. He's also a computer systems consultant, and writes a regular column for REMark.

seem rather expensive at first. But consider that compiling similar information for the IBM computers will cost you about twice that price for *less* information; then the PUP begins to look like a much better value.

The PUP provides some things that IBM doesn't have at any price.

And we'll see that the PUP provides some things that IBM doesn't have at *any* price.

What the PUP contains

Zenith has always been more than generous with their documentation. (Some say it's too extensive and formidable.) The PUP is no exception. When you receive the PUP, you get a 3" binder, a 1½" binder, and five distribution disks. Let's take a look at the contents of the five distribution disks first.

Two of the disks contain the source code for the Z100 version of MS-DOS's basic input/output system (BIOS). A third disk contains the BIOS for the Z150/Z200 computers. All source listings are in assembler and are extremely well commented.

The BIOS is the interface between the computer hardware and the MS-DOS operating system. The BIOS is written by the particular computer manufacturer, such as Zenith or IBM. For Zenith computer systems, IO.SYS is the BIOS; it's

a "hidden" file (hidden from the DIR command).

The fourth disk contains a number of utilities that work on both the Z100 and the IBM-compatible Zenith computers. This disk includes programming utilities (e.g., MASM, CREF, etc.) that are of special importance to assembly language programmers; plus, there are over 20 other utilities that will be of interest to anyone who wants to use some of the advanced features of MS-DOS.

This disk also includes the Basic Screen Editor (BSE). Although I personally prefer S & K Technology's WatchWord, BSE is particularly good for writing programs in assembler.

The last disk is the Keyboard/Font Sources disk. It contains the source code for the keyboard-mapping programs (KEYBXXXX) used by the Z150/Z200. This disk has a total of 24 files, including 22 files for the basic keyboard map. (For example, when assembled UK.ASM will become KEYBUK.COM.) There is also a KEYBXX.OBJ module. And there is a batch file, MAKEMAP.BAT, which is used to assemble and link the compiled program. This will be useful to those who want to get into the low-level programming details of their systems. (Additional information on these files is contained in Chapter 7, discussed below.)

Four out of the five disks are probably not too interesting for those who are not familiar with assembly language and the specifics of the BIOS for the Zenith systems. I think that the PUP documentation and the MS-DOS Utilities disk will be of interest to most people, however.

The MS-DOS Utilities disk may be of enough interest that even non-program-

mers would be willing to buy the PUP for the utilities alone. There are 22 of them in addition to BSE, so it will be worth it to devote a separate article just to the utilities. That is just what I'll do in a future issue of *Sextant*.

In this issue, I'll take a look at the rest of the documentation in some detail. Even those of you who have no intention of programming in assembly language may find the PUP a valuable source of information about how MS-DOS and your computer really work.

The PUP documentation

As I said earlier, the PUP documentation is so extensive that it is divided into two 3-ring binders. The 3" binder, Volume 1, contains all of the information unique to Zenith systems, and is divided into four major sections. I'll review the general contents of each section.

The 1½" binder, Volume 2, contains the documentation on Microsoft's assembly language utilities, such as MASM and CREF.

Volume 1, Part 1: General Information

The first section of the PUP documentation contains two chapters providing an overview of the manual and the MS-DOS system components.

Introduction: Chapter 1 contains ten pages of introductory material related to the organization of the manual and how to use it.

System Components: Chapter 2 contains 30 pages with a general overview of the MS-DOS system components, including the routines stored in read-only memory (ROM), the boot loader, the MS-DOS kernel (MSDOS.SYS), the BIOS (IO.SYS), and the command interpreter (COMMAND.COM). Part of that information is also included in the Zenith MS-DOS manuals; but the PUP manuals include additional information, such as how to modify the BIOS.

Part 2: Standard MS-DOS

This section contains the "heavy-duty" programming information that will be of particular interest to assembly language programmers.

Program Execution: Chapter 3 consists of 21 pages that discuss details of the Program Segment Prefix (PSP), including information on the development of .COM and .EXE programs.

Perhaps the most significant item of interest in this section is the fact that .EXE program files are at least 512 bytes longer than the equivalent .COM file.

The point of that comment is that the amount of memory needed to store an .EXE program file on disk (shown by the DIR command) is *not* the same as the amount of memory it will take up when it's run. Unless the program reserves extra memory for its own use, it will use

up slightly less space while it is running than you would assume from reading the directory listing. That's partly due to the additional header that an .EXE file includes on disk to hold information to be used by the operating system.

(If you load a memory-resident program such as SideKick, therefore, run CHKDSK before and after loading to check the amount of free memory. If you use more than one memory-resident utility, checking the amount of free memory can become critical—since you may not have enough free memory to run another application, such as a word processor.)

System Kernel Features: Chapter 4 contains 299 pages of information required by assembly language programmers on how to use the internal features of MS-DOS. It gives detailed information on the standard system interrupts (INT 20 hexadecimal through INT 2Fh) by which a program can communicate with MS-DOS.

Chapter 4 also goes into considerable detail on all the INT 21h calls—which are used by programs to request standard

This entire section is of particular interest to assembly language programmers.

MS-DOS functions (read a disk, open a file, etc.). I should note in passing that everything covered in this chapter is part of the standard Microsoft DOS interface, so the information is also valid for PC-DOS.

Disk Structures: Chapter 5 contains 38 pages with all of the information that you ever wanted about the structure of data on any floppy or hard disk for any Zenith system.

General information on the standard MS-DOS disk format is included under the discussion of the FORMAT command in the standard Zenith MS-DOS manuals. The PUP contains added information with tables showing the exact disk parameters for each Zenith format.

Information related to the number of directory entries, number of bytes per sector, FAT sectors, directory sectors, and other important information is listed in these tables. This chapter provides information on the exact contents and "decoding" details of the disk directory. It also covers the file allocation table (FAT), which shows exactly where on disk files are stored.

Device Drivers: Chapter 6 has 62 pages with everything you ever wanted to know about how MS-DOS communicates with peripherals and essential elements such

as the screen.

I have mentioned that part of the manual is intended for assembly language programmers; this entire section is a prime example. For those who wish to write their own device drivers, for instance, it includes the special format required for the device-driver headers.

Some of the information in this section is required in order to perform some operations in other languages (e.g., Pascal). You will need to have a modest understanding of assembly language in order to "translate" this information into whatever programming language you are using.

Part 3: ZDS MS-DOS

Foreign Keyboard Remapping: Chapter 7 contains the information necessary to understand object- and source-code files on the Keyboard/Font Sources disk. The fonts on this disk were used in the creation of the standard KEYBXXXX programs (included on the Z150/Z200 MS-DOS disks) that produce the keyboard characters and layouts used in countries outside the U.S.

You can use the information in this chapter to create your own custom fonts for use with the keyboard—for instance, to create additional fonts for countries not included in the Zenith list. This chapter contains 20 pages.

Z100 Software Interface: Chapter 8 is intended for the Z100 systems only; it does not apply to any of the Z138-Z248 computers. In 41 pages, this chapter describes specific details of the MS-DOS BIOS (IO.SYS) interface for the Z100—including interrupts for the keyboard, serial ports, CRT, and other hardware-specific interrupts. BIOS and ROM entry points for programming use are also included here.

Besides the information in this chapter, you will probably need the Z100 Technical Manual (TM-100) if you are programming the Z100. (If you are working with the Z150-series computers, you will probably want to get the Programmer's Reference Manual for the Z-100 PC Series Computers—TM-150. And if you are working with the Z200 computer, you will probably want to get the Z200 PC Series Technical Reference Manual—TM-240.)

In order to understand most of the information in these manuals, some knowledge of assembly language is required.

Part 4: ZDS Utilities

This part of the PUP manual contains two chapters.

BSE: Chapter 9 contains all of the documentation (66 pages) on Zenith's Basic Screen Editor. It's a nice little editor designed and written by Brian Barnes, one of Zenith's MS-DOS experts. Brian wrote it in his "spare time," and it

FTL MODULA-2 \$49.95

The most programming power
for your money

FTL MODULA-2 gives you:

- Full implementation of MODULA-2.
- Split screen, multi-file editor.
- Sources to the run-time modules.
- Complete support for the product.
- 8087 reals available for MSDOS.*

FTL MODULA-2 meets the standards in Niklaus Wirth's book, "Programming in MODULA-2", third edition. No other language gives you better flexibility, code design, or ease of code maintenance.

FTL MODULA-2 gives you compact, rommable code quickly. This is the language of the future. If you program in Pascal or C, FTL MODULA-2 will increase your output per programming hour. If you want to learn high level structured language programming, this is the best starting point.

FTL MODULA-2 was named "product of the year" by Jerry Pournelle in his column, "Computing at Chaos Manor", BYTE Magazine, April, 1986.

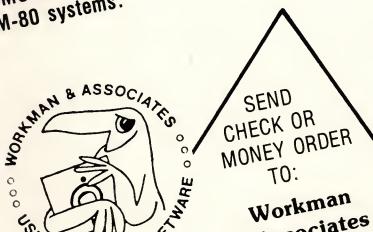
* 8087 support is an addition \$39.95

FTL Editor toolkit \$39.95

The source code to the editor is offered as a toolkit. The editor was written in FTL MODULA-2 and this set of programs provides the novice with working sources to speed up learning. The experienced programmer will find the wealth of pre-written modules a time-saving bonanza.

SAVE \$10.00

Buy both the compiler and the editor toolkit for only \$79.95. That's a \$10.00 savings off the regular price. FTL MODULA-2 is available for both MSDOS and CP/M-80 systems.



SEND
CHECK OR
MONEY ORDER
TO:
**Workman
& Associates**
1925 E. Mountain Street
Pasadena, CA 91104
(818) 791-7979 voice
(818) 791-1013 data M-F 8pm-8am
Or join us on BIX (Byte Information Exchange
WANDA Conference)

MasterCard and VISA welcome.
Please add \$3.00 Shipping and Handling.
Calif. residents please add 6% sales tax.

Circle #229 on Reader Service Card

GET A HEAD START FOR 1987

With these SUPERior Programmer's Packages

The Aspen Systems Subroutine Editor (ASE) you can call from your programs. Design your own screen layouts -- update in several windows simultaneously. ASE is:

- ★ **TRANSPORTABLE** - can be configured for any keyboard and almost any computer or language under MSDOS.
- ★ **VERSATILE** - customize input for any application.
- ★ **COMPLETE** - with full screen edit capabilities and a wide variety of automatic conversions.
- ★ **EASY TO USE** - data and screen layouts described in a single map.
- ★ **FULLY DOCUMENTED** - including examples in BASIC, PASCAL, FORTRAN, and even COBOL.
- ★ **AFFORDABLE** -

ASE is still only.....\$99
DEMO (See what ASE can do for you!).....\$ 3

The Aspen Systems Subroutine Package (ASP) - with many functions unavailable or difficult to perform in high level languages. ASP is:

- ★ **TRANSPORTABLE** - can be configured for almost any computer or language under MSDOS or CP/M.
- ★ **FULLY DOCUMENTED** - a 300 page manual, with examples in BASIC, PASCAL, FORTRAN, and even COBOL.
- ★ **EFFICIENT** - faster, with smaller memory requirements than some high level languages.
- ★ **AFFORDABLE** - ASP is still only.....\$130

ASP includes 150+ subroutines: conversions; sorts; string, bit, date/time functions; decimal arithmetic, and **MORE**.

Prices are PPD (continental USA).
Colorado Residents add 3%.

P. O. Box 1163
Grand Junction, CO 81502
(303) 245-3262
VISA/MasterCard accepted

CP/M and MS-DOS are trademarks of
Digital Research and Microsoft, respectively.



Circle #149 on Reader Service Card

Z-100 AND Z-150 PRODUCTS FROM Computer Graphics Center, Inc.

ADD A NEW DIMENSION TO YOUR H/Z-100

Z3D is a highly versatile, user friendly program for making and displaying three dimensional objects. If you are an artist, architect, engineer or scientist—or have any other interest in graphics—you need Z3D.

Applications of Z3D range from modelling entire towns to demonstrating molecular structures. You are limited only by your imagination and the size of your H/Z100 memory. Features of Z3D include:

- Making objects or scenes in either monochrome or color (up to 36 hues).
- Automatic rotation and scale changes when an eye-point position is selected.
- Automatic shading when a light source position is selected.
- Hidden line removal option.

An illustrated tutorial manual accompanies the program. Only 192K of memory is required and the 8087 coprocessor is optionally supported. Z3D is truly a bargain at its **new price of \$99.00**. Send for your copy today. Please specify your operating system.

Circle Reader Service No. 140 for information on Z3D.

Computer Graphics Center, Inc.

Mastercard 140 UNIVERSITY AVE., SUITE 65 SHOWROOM
and Visa PALO ALTO, CA 94301 444 High St.
Accepted PHONE: 415-325-3111 Palo Alto, CA
California add 6.5% tax, except Santa Clara County add 7.0%.

Circle #140 on Reader Service Card

became so popular at ZDS that it was included in the PUP.

Among other nice features, BSE supports macros; a series of BSE commands can be stored either in memory or in a disk file. That's particularly useful if your work regularly involves a recurring series of commands.

Utilities Guide: Chapter 10 describes the command syntax and some uses for the 22 unique ZDS utility programs. Since the purpose of this article is to give you an overview of the entire PUP, I will discuss the details of these utility programs in a future article.

That article will include each command and its syntax, and some ideas on how these utilities can help you. It's possible that you may find the PUP is worth the price for these utilities alone—even if you never intend to use the programming utilities and documentation.

Part 5: Appendices/Index

The last section of Volume 1 contains five appendices and an index to the contents of the volume.

Appendix A contains 32 pages that provide a *Quick Reference to MS-DOS System Calls* for assembly language programmers. System interrupts (INT 20-2Fh) and MS-DOS function requests for INT 21h are briefly described in this section. It gives such details as the expected register contents upon entry and the

register values upon exit.

Appendix B, *ASCII Symbol Definitions*, contains seven pages of definitions for the ASCII characters, from 0 to 127 decimal. On the last page of this appendix is a table showing the graphics characters that the ROM of the '150 (and Zenith's other IBM compatibles) will display.

Appendix C, *Z100 Escape Sequences*, consists of ten pages giving the escape sequences that are required for Z100 screen control or are generated by the various Z100 function keys. (This information is documented in more detail in the Z100 Technical Manual.)

Appendix D is the ten-page *Glossary*. Although it seems to be fairly complete, I think that it needs a few additions. For example, both "paragraph" and "page" are defined, but "word" is not.

That may seem to be nit-picking, but remember that many who are using the glossary will be unfamiliar with the "back-words" storage of values. Defining "word" would be a logical place to discuss the fact that bytes are reversed; so when you see 99 88 in the hex dump of a listing, the value will actually be 88 99.

(Based on my experience with the *FlipFast* books, though, I know it is difficult to develop a good glossary. It's much easier to be critical of something than to develop it in the first place.)

The last appendix, Appendix E, contains 12 pages of *Programming Hints*

with sections on interrupts, system calls, device management, memory management, process management, file and directory management, and a section on miscellaneous topics. These hints are directed specifically toward assembly language programmers and are of little, if any, value in other languages (particularly BASIC).

The 23-page *Index* rounds out the last part of Volume 1. The index is comprehensive, and I have found it to be quite helpful in locating information. Perhaps the only limitation is that you must know the internal MS-DOS terminology, since many of the items are specifically indexed that way. But I think that is a reasonable assumption given the purpose of the PUP.

Volume 2: Programmer's Utilities

From what I have seen, Volume 2 contains most of the standard information that is provided in Microsoft's Macro Assembler (MASM) package. Volume 2 includes chapters with information on MASM, LINK, SYMDEB, CREF, LIB, and MAKE. There is also an introductory chapter.

Part 6: User's Guide

The *User's Guide* contains specific information about the syntax and usage of the Microsoft assembly language programmer's utilities that are part of the

TaxAide

Income tax preparation worksheets for use with your Lotus 1-2-3.

- PRODUCES IRS APPROVED PRINTOUTS—NO SPECIAL FORMS TO BUY •
 - Easy to Use — Menu Driven — Integrated •
 - Screen Displays Resemble Actual IRS forms •
 - Automatically Calculates Income Tax •
 - Full Technical Support — Satisfaction Guaranteed •

TaxAide Plus — Provides worksheets for Forms 1040, 1040A, 2106, 2119, 2210, 2441, 3468, 3800, 3903, 4562, and 6251; Schedules 1, A, B, C, D, E, F, G, R, SE, and W; plus IRS worksheets. Multiple Forms 2106 and 4562 and Schedules C and SE. TAXAIDE PLUS PRODUCES IRS APPROVED PRINTOUTS FOR ALL FORMS AND SCHEDULES, INCLUDING 1040 AND 1040A. Ideal for income tax preparers. **\$59.95**

TaxAide Personal — Provides worksheets for Forms 1040, 1040A, 2106, 2441, and Schedules 1, A, B, D, G, R, and W, plus IRS worksheets. Two Forms 2106 are provided. Produces IRS approved printouts (except Forms 1040 and 1040A). **\$29.95**

REQUIRES ONLY 192K of RAM — If you have more memory, you may still run your background program such as SideKick or Perks.

Requires Lotus 1-2-3 Release 1A or later, and DOS 2.0 or later.

To order, send check to:
Kansas residents add
5% sales tax

Software Applications of Wichita
2204 Winstead Circle
Wichita, KS 67226

For VISA or MasterCard orders, call (316) 684-0304.

TaxAide is a trademark of Software Applications of Wichita
Lotus and 1-2-3 are registered trademarks of Lotus Development Corp

SideKick is a trademark of Belland International, Inc
Perks is a trademark of Barry A. Watzman

Circle #190 on Reader Service Card

HOME FINANCE SYSTEM VERSION 2

—An extensive Home Finance System that keeps track of checking, asset accounts (cash, savings, IRAs, CDs), and regular bill payments. Let your printer write your checks for you on **any** business-sized check (design your own check format).

—Checks have user defined codes and a separate flag for tax deductible items.

—Many reports, including listing all checks, or checks by codes or tax flag.

—System consists of 130 page users manual with 5 program disks (5-1/4") and a sample data disk.

Hardware: H8/HZ89 (64K) or HZ100 with 2 disk drives. Any Heath®, Zenith® or other printer.

Software: CP/M or CP/M-85/86 (Ver. 2.2) and MBASIC 5.21 for CP/M.

Order: Complete System \$89† (specify hard or soft sector 5 1/4", HZ89 or HZ100). Manual alone \$21.†

MasterCard/Visa accepted, please include your phone number.

†Prices include shipping.



**Jay H. Gold, M.D.
Jay Gold Software
Box 2024, Des Moines, IA 50310
(515) 279-9821**

Circle #120 on Reader Service Card

package.

Introduction: Chapter 1 is, as usual, a short (four-page) introduction with information on notation conventions used in the volume.

MASM: A Macro Assembler: Chapter 2 contains 24 pages on the use of MASM, the macro assembler. Additional information on MASM is included in Part 7 of the PUP manual.

LINK: A Linker: Chapter 3 contains 32 pages on the use of LINK. The PUP's documentation for LINK contains more information than is included in the manual for standard MS-DOS.

SYMDEB: A Symbolic DEBUG Utility: Chapter 4 contains 72 pages on the use of SYMDEB, Microsoft's symbolic debugger. Like MS-DOS's standard DEBUG utility, it lets you display and edit executable files. In order to fully utilize the features of the symbolic debugger, you *must* have the source code for the program (<FILENAME>.ASM). First, you have to create a special symbol table (using the MAPSYM utility included in the PUP). Then SYMDEB can display the code in terms of the symbols that your program has assigned to expressions (rather than just giving the memory locations at which they are stored).

If you program in assembly language, I think that a symbolic debugger is essential. Also, SYMDEB has a number of unique subcommands in addition to the

ones provided by DEBUG.

SYMDEB has a help function (the ? command—same as with the Z150/Z200 series ROM). Also, SYMDEB allows you to use a *limited form* of I/O redirection—something that you can't do with DEBUG. Input and output redirection are limited to the use of the console (CON—the default input device) and the first serial port (COM1). In addition to the standard I/O redirection < and > symbols, you can also use the equal sign (=) to redirect both input and output to CON or COM1.

The use of output redirection (>) is fairly obvious—you may want to print a debugging session as you perform various actions with some commands.

The standard input device is defined as the console (the keyboard and CRT); so, all SYMDEB commands are normally entered through the keyboard. The use of input redirection (<, to get input from a disk file, say) is not obvious, and I have not been able to develop a "normal" use for either it or the equal sign.

Since a printer is clearly not an input device, it is not possible to redirect both input and output when COM1 is a printer. I suppose you could connect a terminal (i.e., a CRT device) to COM1, but that does not make much sense, and such examples tend to be artificial.

If you have a reasonable use for input redirection (<) or both input and output

(=) redirection, you might want to write a letter to me or to *Sextant* and share that information.

(Redirection is an important element in a number of the PUP's utilities, and will be addressed in the article on the utilities.)

Another step up from DEBUG is SYMDEB's G (Go) command, which executes a program up to a specified breakpoint. In addition to that, SYMDEB allows you to directly set program breakpoints (BP), as well as clear (BC), disable (BD), enable (BE), and list (BL) them.

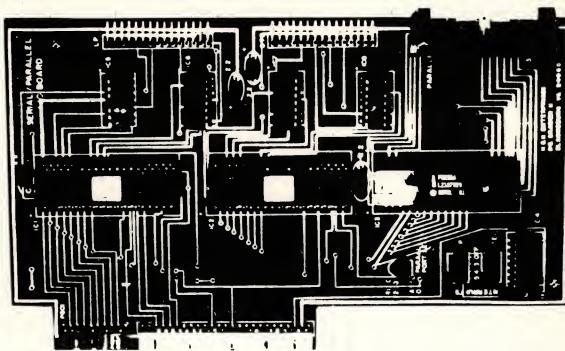
In addition to the standard D (Dump) subcommand, you can also dump ASCII (DA), bytes (DB), words (DW), and double-words (DD). And for those of you with lots of math in your programs, you can dump in terms of short reals (DS), long reals (DL), and ten-byte reals (DT).

Since SYMDEB allows you to work with symbols, two commands are provided for that purpose. You can examine the symbol map (X), and open the symbol map (XO).

CREF: A Cross-Reference Utility: Chapter 5 contains nine pages of information on the CREF program, which is invoked during program assembly when you ask MASM to produce a cross-reference file listing all the symbols used in the program.

LIB: A Library Manager: Chapter 6 contains 15 pages of information on the

THE ORIGINAL SERIAL/PARALLEL INTERFACE BOARD



I/O Board

Replace the H88-3 to upgrade your system or use the KC 1222 in place of it! Providing the two standard serial ports PLUS three parallel ports on one card, you have the best of both worlds! Use the parallel ports for printers, joystick, instrumentation/control... KC 1222 has components including cables and hardware. KP 1222 has the same items less the serial chips and serial cables.

KC 1222 Complete Kit	\$128
KP 1222 Parallel Kit	97
PC 1222 Bareboard	48.50

Printer Drivers

The C. Itoh, Epson, Okidatas, Gemini and others are supported under HDOS and Heath's CP/M \$20 Magnolia Patch \$10

Printer Packages

Our printer packages are fully checked out and set up for your system. All you need to do is plug it in and start using it! No more fussing with switch settings, making cables... the aggravation is gone! The package includes cable, assembled and tested, set up printer and your choice of HDOS or CP/M software.

Dot Matrix Packages Include Cable and Software
Gemini 10x \$350
Prowriter 8510AP+ \$395
NEW Prowriter 8510 SP (180cps) 595
NEW Prowriter 8510 BPI (IBM compatible) 480
NEW Prowriter 8510 SCP (color 180 cps) 690

Daisywheel Packages Include
Cable And Software
Silver Reed EXP 500 \$500
Silver Reed EXP 550 720
NEW Silver Reed 400 395
C. Itoh A10-20 595

Z-100/Z-150 Folks: Subtract \$20... you don't need the software!

ZDS HARDWARE • PRINTER PACKAGES • SOFTWARE

MATHGEN: The keyboard math interpreter brings mathematics to life! Function roots, max and min, evaluate function integrals and first derivatives, hyperbolics, exponential forms, logs, trig functions, factorials, the list goes on! Define the functions from a file, batch processing style, or from the keyboard. Plot to the screen or your printer. CP/M or HDOS for only \$40.00. Now for Z-Dos, too!

DATAPLOT: Using the dot addressable graphics of your Prowriter, plot any definable math function in single or double density. Select axis size and scale, plotting quadrant, etc. Generate your data by MATHGEN, MBASIC or any other language. CP/M or HDOS for \$25.00.

PRN: A printer utility for the Prowriter, Epsons, and H14. Print headers, footers, pause at page end, print multiple copies, left margin, printer mode and a host of other functions. CP/M or HDOS for \$25.00.

JS: Use your parallel ports for REAL TIME joysticks! Use up to four at a time with your favorite MBASIC or machine language keyboard game for HDOS. Program the response speed of the sticks, key sequence for different games, etc. Uses Atari joysticks. Only \$25.00

We fully support all of our independent-vendor products. If you have a question, we will **never** refer you to someone else.

NEW PRODUCTS

MODEMS	BUFFERS
Anchor Signalman Mark I	\$95
Anchor Signalman Mark XII	\$350
Novation D-Cat	\$189
Novation Smart-Cat 103/212	\$550
Volksmodem	\$75
TTX 16K	\$160
TTX 64K	250
Quadram 16K	175

SURGE SUPPRESSORS

EPD: "Lime"	\$79
EPD: "Orange"	120
Panamax SS4LCS	95
Panamax Maxi Strp	35

SWEET P-6-SHOOTER PLOTTER \$895

As authorized ZDS dealers, we support the Heath/Zenith community with competitive prices and post sales support. Contact us for price and delivery on the Z system and accessories of your dreams.

VA residents add 4% tax. Please add 3% or \$2.00 minimum for shipping. Foreign orders in U.S. dollars or drawn on a U.S. bank. Also include customs house and address.

Group purchase discounts are available—contact us for details.

H & H Computer Enterprises, Inc.

P.O. Drawer H Blacksburg, VA 24060

(703) 552-0599



LIB program that is used to manage a library of object modules that can be used with LINK.

MAKE: A Program Maintainer: Chapter 7 contains six pages on the use of MAKE, the Microsoft program that helps you update programs. It is, in many ways, quite similar to the Zenith DO utility, which I will cover in the utilities discussion in a future issue. (I think that the DO utility is much easier to use, and I also like the syntax and control file construction better.)

Appendix A contains 19 pages of error messages, listed alphabetically—including the error messages for MASM and LINK. For some reason, however, there is no error-message listing anywhere in this section for CREF and LIB. I guess that Zenith assumes that you will be able to decode some of Microsoft's incredibly cryptic error messages.

The *Index* to the first six parts takes up the last 11 pages of the 3" binder. Based on my use, it appears to be complete; but you need to understand assembly language terminology before you can find anything. Again, it's not unreasonable to assume that the reader of this section has that familiarity with assembly language.

Part 7: MASM Reference Manual

This part of the manual includes information on the use of the macro assembler. This section is definitely *not* for the uninitiated. Since the use of its information is highly specialized, I will just include a list of the chapters and their general contents.

Introduction: Chapter 1 is a short three pages giving an overview and describing the notations used.

Elements of the Assembler: Chapter 2 contains 11 pages on elements of the assembler—including such things as the character set, reserved words, and similar information of a general nature.

Program Structure: Chapter 3 discusses the structure of an assembly language program, and includes information on assembler directives (16 pages).

Types and Declarations: Chapter 4 contains 20 pages on the declaration of variables and types of variables used in assembly language, as well as related assembler directives.

Operands and Expressions: Chapter 5 is a comparatively long 31 pages on the use of operands and expressions.

Global Declarations: Chapter 6 talks about global declarations in a short seven pages.

Conditional Assembly: Chapter 7 contains information on the use of directives for conditional assembly (in six pages).

Macro Directives: Chapter 8 contains 15 pages on various directives that can be used with MASM.

File Control: Chapter 9 contains information about the use of the MASM file control directives (in 11 pages).

Appendix A, *Instruction Summary*, in-

cludes a complete summary list of the instructions supported by the microprocessors that MS-DOS runs on: 8086 instructions, 8087 instruction mnemonics, 80186 instruction mnemonics, 80286 non-protected instruction mnemonics, 80286 protected instruction mnemonics, and 80287 instruction mnemonics (all in 15 pages).

Appendix B, *Directive Summary*, includes five pages of valid MASM directives.

Appendix C, *Segment Names for High-Level Languages*, has nine pages of information on segment names that Microsoft uses in its high-level language packages (i.e., C, FORTRAN, and Pascal).

And Part 7 ends with its own 11-page *Index*.

Is it worth \$225?

Even though the PUP appears to be expensive at \$225, you get a lot of information about the technical details of your operating system, as well as information about the BIOS used by each of the Ze-

Did you ever go to an IBM dealer and ask for BIOS information?

nith systems. The technical information, BIOS listings on disk, Zenith utility programs, and the Microsoft assembly language utilities are all included in the \$225 price.

Dare to compare with IBM?

Why not? We should not neglect the opportunity to compare Zenith's PUP to what is available to those people who decided to buy their computer from the world's largest manufacturer of Zenith-compatible computers.

But as I started to do this comparison, I found that it was not as easy as I thought. The information is scattered all over the place. And did you ever go to an IBM dealer and ask for BIOS information? They look at you as though you have some kind of disease or something—at least the local dealers did. But I finally found all of the information at the last place I looked. (Odd how it always seems to work out that way.)

In any case, my idea was to try to duplicate the information that could be found in the PUP, and here is the result.

First, you will need to get the DOS Technical Reference. I found that the Version 3.10 manual sells for \$48 at a local IBM dealer. Next, you will need to get the BIOS information. You can get the IBM PC Technical Reference for \$36, the XT Technical Reference for \$53, and the AT Technical Reference for \$105. So, depending on whether you needed the

information for all three machines, you could have already spent as much as \$242.

My research indicates that the AT Technical Reference does not include any listing of the BIOS source code; the PC and XT manuals do, but no disk is included with the source code. If you own one of those systems, and if you plan to make any modifications, you will undoubtedly have lots of fun typing in all of that assembly language code, not to mention trying to debug it.

I guess we will also need to get the assembler and supporting utilities, so it looks like another \$150 down the drain. The IBM dealer that I visited had both the \$150 Macro Assembler package from Microsoft and the "official" package from IBM. The IBM package was \$5 more, although I did not see any difference except for the name on the box.

In round numbers, we have already blown away over \$400 by the time sales tax is figured, and we still don't have any of the utilities that Zenith provides.

Since nothing like that utility set is available from either IBM or Microsoft, I guess we'll have to be content with our \$400 expenditure—unless we want to get a good compiler and supporting utilities so that we can write similar programs. Depending on the compiler and the utility library, we could spend as much as \$600 more for the combination—not to mention the time required to write and debug the programs. And all of this obviously assumes that we know enough assembler and/or C to write the programs in the first place.

Last, but not least. . .

I have always found it instructive to compare Zenith with IBM. I have 20 years' experience in mainframe computer systems, primarily IBM, and I am continually amazed by the ingenuity of Heath and Zenith systems. It is difficult in many cases to fully appreciate Zenith operating-system software until you do a direct comparison with IBM. Perhaps we should do more of that.

I am always glad to answer questions about my articles and books if you will enclose a stamped, self-addressed envelope (preferably a #10 business-size envelope) with your letter. And I am always interested in hearing about subjects that you would like me to write about in *Sextant*.

William M. Adney
P.O. Box 531655
Grand Prairie, TX 75053-1655

Ordering Information

MS-DOS Version 3 Programmer's Utility Pack, model #CB-3163-30, \$225.
Heath Company
Benton Harbor, MI 49022
800/253-0570; in Michigan and Alaska,
616/982-3411

I want to subscribe to *Buss!*

Send me:

- 20 issues for \$28 (\$40 overseas)
- 10 issues for \$18 (\$24 overseas)

I've enclosed payment. Rush me *The Buss Directory!* (checks must be in U.S. dollars payable on a U.S. bank)

Bill me. I'll receive *The Buss Directory* when I send my payment.
 Charge my Visa MasterCard

Card # _____

Expiration date _____

Name _____

Address _____



Buss, Dept. S26
716 E Street, S.E.
Washington, DC 20003

S26-B

I want to subscribe to *Buss!*

Send me:

- 20 issues for \$28 (\$40 overseas)
- 10 issues for \$18 (\$24 overseas)

I've enclosed payment. Rush me *The Buss Directory!* (checks must be in U.S. dollars payable on a U.S. bank)

Bill me. I'll receive *The Buss Directory* when I send my payment.
 Charge my Visa MasterCard

Card # _____

Expiration date _____

Name _____

Address _____



Buss, Dept. S26
716 E Street, S.E.
Washington, DC 20003

S26-B

When mailing in an envelope,
please use ZIP code 20003.



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 13098 WASHINGTON, D.C.

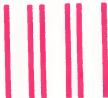
POSTAGE WILL BE PAID BY ADDRESSEE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

A vertical column of ten thick, black, horizontal bars used for postal sorting.

BUSS/SEXTANT
716 E Street, S.E.
Washington, DC 20077-0860

When mailing in an envelope,
please use ZIP code 20003.

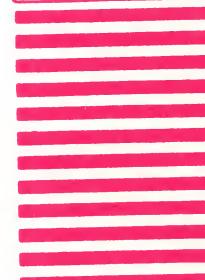


BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 13098 WASHINGTON, D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

A vertical column of ten thick, black, horizontal bars used for postal sorting.

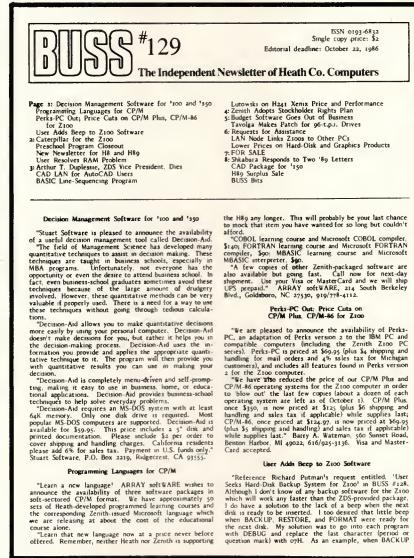
BUSS/SEXTANT
716 E Street, S.E.
Washington, DC 20077-0860

**Look what happens when we put
our minds together!**

Great ideas come together in *Buss* newsletter. Subscribe, and you'll get inside information from the Heath/Zenith world. Since 1977, *Buss* has been a reliable news source and a dynamic forum between users and vendors.

No other Heath/Zenith-specific publication comes to you as often as *Buss*. Twenty times a year (every two to three weeks), you'll be on top of events and information about your Heath/Zenith computer.

Buss costs only \$28 for a year's subscription—a small amount compared to the wealth of news and information you'll get to support your system.



Here's what *Buss* offers:

- New product sources and upgrades for all machines, from the '8 to the '248
- Users' advice and discoveries about their systems
- FREE "For Sale" ads so you can buy or sell equipment
- "Requests for Assistance"—get help for your system through the network of 6,000 *Buss* subscribers
- News and events in the Heath/Zenith community
- Unbiased coverage of Heath/Zenith systems

BUSS

The Independent Newsletter of Heath/Zenith Computers

**GET 20 ISSUES FOR \$28.
CALL NOW, TOLL FREE:
800/341-1522**

Charge it to your Visa or MasterCard
DATATEL 800™ for orders only,
Monday-Friday 8 a.m.-9 p.m., Saturday
9 a.m.-5 p.m. Eastern time

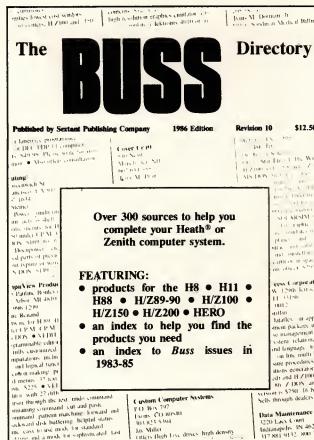
Or mail coupon with payment now to:
Buss, Dept. S26
716 E Street, S.E.
Washington, DC 20003

**Subscribe now and get this
FREE BONUS!**

The Buss Directory is an annual guide to over 300 Heath/Zenith vendors of hardware and software products, as well as a handy index to *Buss* articles from 1983 to mid-1986.

You'll have up-to-date information at your fingertips about available resources for your system. It's an indispensable reference manual.

Pay now and get this directory, a \$12.50 value, *FREE!*



Count me in—start my *Buss* subscription!

Send me:

20 issues for \$28 (\$40 overseas)
 10 issues for \$18 (\$24 overseas)

Name _____

Address

Buss, Dept. S26, 716 E Street, S.E., Washington, DC 20003

The H89 SPEED Center

4MHz mod

\$45

An easy to install plug-in module. No trace cutting or soldering. Speed may be toggled with software. Includes a replacement Z80A (4MHz). Includes CP/M software support for Heath, CDR Systems and Magnolia. Call or write for info on HDOS support. Specify disk format.

6MHz mod

\$59

Similar to our 4MHz modification, but increases the CPU speed to 6MHz. Requires some soldering on the CPU board. Includes a replacement Z80B (6MHz). May require replacing additional parts. Some technical knowledge is recommended for installation. CP/M support only. Specify disk format.

CDR Super RAM 89

Make your H89 a speed whiz with this board and the included RAM drive software. With our 6MHz mod and the RAM drive, the Heath BIOS assembles (using MAKEBIOS) in less than 1 1/2 minutes! Note our discount prices!

Main board w/o RAM - specify disk format	\$175
Expansion board w/o RAM	\$69
Main board w/512K RAM	\$239
System w/MEGABYTE RAM	\$369
Clock option	\$45
SCSI option	\$85

REP3 - Automatic Key Repeat

Stop wasting time - make full use of the repeat function just like the Z100. Simple plug-in installation on your keyboard assembly. Hold any key down for half a second and the key begins repeating. Combine this with our 4MHz mod and make WordStar fly! Provision for a defeat switch.

Kit	\$35
Assembled	\$45

WSPATCH

\$20

Adds H19/H89 function key patches to versions 3.0 or 3.3 of WordStar. Key functions similar to the PIE editor. Includes provision for redefining the keys by the user. Also includes a printer driver for the Epson MX80/FX80 printers.

EMULATE

Allows the H89 to read/write to the following disk formats.

Actrix	Eagle II	Morrow MD	Superbrain Jr.
AMPRO	Epson QX-10	NCR DecMate 5	Televideo
Beehive Tpr	Fujitsu CP/M86	NEC PC-8001A	TRS80-1 CP/M
CDR	IBM CP/M86	Osborne 1	TRS80-3 CP/M
Cromemco	IMS 5000	Otrona	TRS80-4 CP/M
DEC VT180	Kaypro II	PMC MicroMate	Visual 1050
DEC Rainbow	Magnolia	Royal/Adler	Xerox 820
		Sanyo 1100	Zorba

Now includes 44 formats! Uses a modified BIOS which is included with the program. Allows reading of 40-track disks in an 80-track drive.

For H37 with Heath CP/M	\$59
For CDR BIOS 2.91	\$49
Check for Magnolia version	

ANALYTICAL PRODUCTS

805/688-0826

213 Teri Sue Lane

Buellton, CA 93427

CDR Controllers

Double density disk controllers. Includes board, instruction manuals, ROMs and software. Allows control of 5 and 8 inch drives. Includes BIOS source code and some useful utilities. Please include your CP/M s/n when ordering.

For the H89 FDC-880H **\$349**

ZCPR3

We are licensed by ECHELON, Inc. to distribute the Z-System for the Heath market. This includes the ZCPR3 system with supporting software utilities, the ZRDOS-plus BDOS replacement and applicable documentation. Comes already installed and includes a bootable disk ready to run.

ZCPR Gives you many features found in UNIX and MSDOS. The ZRDOS is not in the public domain and gives you automatic logon of disks. No more "BDOS ERROR ON B:" messages just because you forgot to do a ^ C after changing disks.

Available for Heath BIOS 2.2.03 and .04 and for CDR BIOS. Check for availability of other BIOS configurations. Call or write for a brochure on this excellent product.

ZCPR3 - specify format and hardware **\$98**

TIM2 - Real Time Clock

This circuit installs in the left hand expansion slots of the H89 or H89A. Can provide application programs with time and date information. Includes battery backup for continuous timekeeping when the computer is turned off. Program access to the clock is through an I/O port. The port address is user selectable by means of a jumper. Requires soldering 4 wires to the CPU board. A prepared ribbon cable is included.

Kit	\$65
Assembled	\$75
Software on disk - specify disk format	\$10

DATESTAMPER

Product of Plu*Perfect Systems. Provides automatic time and date stamping for CP/M 2.2 files. Works with many real time clocks, including our own TIM2 product.

CP/M - specify disk format **\$45**

Specify Disk Format on Software

CALL OR WRITE FOR CATALOG PRICES SUBJECT TO CHANGE

Terms: Check or Money Order - VISA/MC - C.O.D.

Add \$3 per order for shipping and handling

California residents add 6% tax

The Eight-Bit World

Walter J. Janowski

Hectic times

HUGCON 5

HDOS 3.0—here at last?

Lightning strikes

Survival

The basic survival system

Survival extras

United we stand

The quest for the perfect editor

Volunteer needed

Write!

Hectic times

The past two months have been busy ones. Between HUGCON, lightning storms, and wrapping up the kitchen/bathroom remodeling, I've barely had time to breathe, let alone write a column.

Some of the happenings of the past months have caused me to reflect on the subject of surviving in the eight-bit world. Forgive me if I postpone the subjects I may have promised for this column, and instead spend a little time discussing some survival techniques.

HUGCON 5

Way back at HUGCON 1, there was a special sense of excitement that went into the idea of the users of Heath/Zenith systems getting together for an annual gathering. There was still an air of mystery surrounding microcomputers; the Z100 series was just being introduced; and Heath/Zenith management thought we were an important part of their customer base.

This August brought about the fifth annual gathering, and I couldn't help feeling that the magic has gone.

Somehow it just doesn't seem as special to get together any more. Maybe it's just the accessibility of computers today, the way everybody and their brother has one.

The folks at HUG certainly aren't to blame. They gave their all, as usual.

(Unfortunately, Zenith Data Systems turned out to be a no-show at the last minute. It seems their coordinator changed jobs a couple of weeks before the conference, and everything fell through the cracks. The Heath Company booth was alive and well, though. And Heath's sales room had the bargain prices they've always had at HUGCON.)

Representation of the eight-bit world also reached a new all-time low. Newline Software, for instance, marked all their H89 software at half price—and announced that when supplies were gone there would be no more.

Yes, C.D.R., Kres, The Software Toolworks, and a few others still had eight-bit products to sell. But new products were few and far between.

One company keeping things active is Clay Montgomery's SigmaSoft and Systems. While my experience with them is limited, they've always drawn good comments. This time, they showed their new computer-aided design program. It's designed to support their graphics board, which has been out for a little while now.

(HDOS loyalists will be pleased to hear that Darron Shaffer, SigmaSoft's programmer, finds it's easiest to do all the development in HDOS, then transport it over to CP/M.)

Another bright spot was TMSI (Technical Micro Systems, Inc.), who showed their H-1000 co-processor board and Write-Hand-Man. They also displayed a handful of *new* products—including a flicker-reducing modification for the H19 terminal (and H89s), a set of low-power replacement chips to reduce power consumption, and an enhanced character-generator ROM for the H19. (I'll be reviewing the "Flicker Free Kit" in the next few months.) Keep an eye on TMSI for pumping a little new life into the eight-bit world.

Lee Hart of TMSI told me this was his most profitable show ever, which seems to indicate that user interest in the eight-bit world is still very much alive.

HDOS 3.0—here at last?

At HUGCON 1, the running joke was "Where's HDOS 3.0?" At HUGCON 2, Heath/Zenith announced that their HUG user base was important enough to justify development of HDOS 3.0. At HUGCON 3, we heard from Bill Parrott who was writing HDOS 3.0; he gave us a few tantalizing insights into what it might contain. By HUGCON 4, rumors flowed that Zenith was pulling funding for development; some people had questions about whether or not HUG could produce HDOS 3.0 themselves if given the chance. It looked very much as though chances of there ever being an HDOS 3.0 were fading away.

Then came HUGCON 5 where, at an innocent-enough-looking session on software questions and answers, Bill Parrott dropped the bombshell. HDOS 3.0 was done, in his possession, and about to be handed over to Bob Ellerton of HUG that very weekend.

Bill and I later discussed what we saw as the possible marketing options for HUG. We felt most faithful HDOS'ers would be willing to cough up as much as \$50 to \$100 for HDOS 3.0, but we wondered if there were enough of us left to justify the effort. Of course, HUG could always give it away. . . .

Then, at the dinner Saturday night, Bob Ellerton announced that HUG would place HDOS 3.0 into the public domain!

Initial plans were to place it on the bulletin board at the HUG office (616/982-3956) and make it available for free

downloading. HDOS 3.0 is somewhere around half a megabyte in size; but downloading it seems no more unreasonable than downloading the complete set of files for ZCPR while paying connect time. Many people have done that from HUGSIG on CompuServe. (Maybe local HUG libraries will be able to download HDOS 3.0 and then make it available locally on disk.)

According to Bill Parrott, HDOS 3.0 is an ORG 0 disk operating system. That is, it resides in memory beginning at memory location zero; as such, it is a true 64K system. (No more "System Has 56K of Memory", when you know darn well it has more.) It will allow a much larger number of device drivers, and it will support batch files similar to MS-DOS.

(Ever notice how similar MS-DOS has already become to HDOS? Not too surprising, since MS-DOS's guru is J. Gordon Letwin, the original author of HDOS. It's always seemed as if he had his eye on more powerful operating systems such as Unix, Digital Equipment Corp.'s operating system, when he was writing HDOS. His influence is finally coming through in MS-DOS.)

[In November, some of the Sextant staff attended the Western Regional HUG Conference, where Bill Parrott had an updated copy of HDOS 3. Bill said he'll mail a copy to anyone who sends him \$25 at 7010 Caenen Ave., Shawnee, KS 66216. Specify H17, H37, or H47 disk format—you'll get three to seven disks.]

Lightning strikes

A few weeks ago, I was out of town on a business trip. My wife called me at my hotel to tell me that there had been a lightning storm the previous night—and that the telephones were unable to accept any incoming calls. Knowing the phone company's current policy of charging for repair calls not related to their equipment, I suggested she unplug each of our telephones until the problem went away. She later phoned back to say that none of the phones seemed to be at fault.

Upon returning home a few days later, I tried to analyze the problem. What could be the cause? While walking through my office, my eyes fell upon the only other item connected to the phone lines: my modem. With a sinking feeling in the pit of my stomach, I unplugged the modem. The problem went away.

The modem cost approximately \$250. A quick call to my insurance company verified that the deductible on my homeowner's policy was \$250. Sigh.

My modem is (or was) a Prometheus ProModem 1200. It has performed flawlessly for over two years. With a few calls to Prometheus and a little dedicated troubleshooting, I'm sure I should be able to bring the poor thing back to life; but I don't know when I'll be able to spare the time. Until I do, I'll have to wait to track down HDOS 3.0.

Don't be surprised if you can't reach

me on the boards.

(Late update: It appears that my modem will function normally when connected, but it does still seem to load down the phone lines when it's off. So I can still communicate on a somewhat limited basis. As I'm writing this, though, I just noticed a few flashes of lightning out the window. Keep your fingers crossed.)

Survival

With the current state of the eight-bit marketplace, how can we hope to best survive with our old workhorses?

With a little effort and logical organization, I think we can do it. Let's explore the possibilities a bit.

Let me start with a definition or two. By "surviving," I mean continuing to utilize our eight-bit machines for a useful and fulfilling future. This includes maintaining current standards as well as continuing to grow.

There are probably a number of you out there with just a basic H89 with one hard-sectored drive. If so, that may be all you need or can afford. Unfortunately, those of you just idly tinkering with your machines—or too financially strapped to invest in them—may find it difficult to keep up.

(But people can't really complain about the lack of new word processors, say, when they're writing the letter with CP/M's ED or HDOS's EDIT. That sort of complaint does tend to irk me. Some users haven't spent a penny on their system since 1979, but can't understand why vendors don't want to make an effort to support them.)

So let's start by defining our basic survival system. I'll start with a basic one-drive H89 and explore what add-ons are necessary for surviving.

(I apologize to H8 users, but my familiarity with that machine is rather limited. Most of the same principles apply, however, although you may need to acquire a different product to achieve the same end.)

The basic survival system

First of all, if you're running just HDOS, get a copy of CP/M right away (no matter how tough it is to admit that you need CP/M)!

I expect a brief flurry of HDOS activity after the release of version 3.0; nonetheless, I think HDOS will have to be left for your private hacking or for running those old favorite programs you already have. External vendor support for HDOS will soon be close to, even if not totally, nonexistent.

Also, if you don't snatch up a copy of CP/M now, you may never have a chance. You may not even be able to find it at all, other than at a close-out sale.

Second, that old hard-sector controller just won't cut it. Hard-sectored diskettes are getting more expensive and harder to find. Not only that, but you eliminate the possibility of swapping disks between

yours and a "foreign" machine. (More on this later.) Heath no longer manufactures the Z37 soft-sector controller card, but look to TMSI, C.D.R., or Magnolia for alternatives.

Also, you'll need more than that loyal old single-sided internal drive. A pair of soft-sector double-sided 48-track drives is a must. Add-on kits are available from several vendors, including two-drive kits that can be installed internally in the old single-drive slot.

You can get double the disk capacity with 96-track drives; but there, too, you may sacrifice compatibility with "foreign" machines.

If you're running under HDOS, of course, it should be fairly easy to run both 48- and 96-track drives. Use 96-track for regular operation; keep 48-track on hand for compatibility and file transfer. It's just a matter of having both controller boards and the appropriate device drivers.

Those of you running under CP/M will need a custom BIOS to handle both types of drives at the same time, but that's not out of the question. (You might even want to keep HDOS and the old 48-track drives to use them just as a transfer medium. To the best of my knowledge, however, there are no HDOS-CP/M transfer programs using soft-sectored format. I could be wrong, but apparently the only HDOS-CP/M file-transfer programs are from J. J. Thompson, and they require hard-sector format.)

In any event, once you've settled on disk operation, the next most important aspect of survival will be enabling your machine to communicate with the outside world.

If you don't have one, add a three-port serial card or equivalent to allow you to connect a printer, modem, etc., to your system. Again, the unit from Heath is no longer available; but browse the ads in *Sextant* for one of several alternatives. It is important to consider any machine-specific hardware add-ons now; the future of their availability is somewhat shaky.

If you feel competent to make the evaluation, don't ignore the possibility of purchasing used equipment. (Local HUGs are a great source, for instance, and there are "For Sale" sections in both *Buss* and *H-Scoop* newsletters.)

There is a lot of support available in the user community for eight-bit machines, but you have to be able to get to it. If you haven't already, take the plunge and buy a modem. (See some of my previous columns for discussions of various pieces of telecommunications software.)

Once you have a modem, you'll open up channels of communication with other users—as well as gain access to the mass of public domain software available. The modem may well be your most valuable survival tool. You and your system will never feel lonely and neglected again.

CP/M may have gained us some degree of machine compatibility, but as an oper-

ating system it leaves a lot to be desired. To take some of the pain out of it, invest in ZCPR, the public domain enhancement to CP/M; it is available in do-it-yourself form from various bulletin boards, or pre-configured for the H/Z machines from Analytical Products.

ZCPR is marketed by Echelon, Inc. Be sure to get a subscription to their *Z-News* newsletter to stay in touch with ZCPR and other eight-bit happenings.

Survival extras

Once you have a full system, there are a few more add-ons that, while not absolutely necessary, are quite useful for survival.

First of all, stop chugging along with a 2-megahertz system clock. There are several 4- and 6-MHz modification kits around, and they're easy to install. It may not seem like much, but once you've speeded up your system, you'll never be able to go back.

I've warned against waiting for close-outs to buy that software you've always wanted; but that doesn't mean you shouldn't take advantage of close-outs when they come along. As a matter of fact, the low prices of CP/M software are one of the best advantages of an eight-bit system.

But who says you have to limit yourself to software produced for Heath/Zenith machines only?

With programs like COMPAT from Workman & Associates, EMULATE from Analytical Products, and MODIFY 89 from C.D.R. Systems, you can read and write disks for a large number of other machines—including Z150 MS-DOS compatibles. Most commercial eight-bit software was distributed in 48-track format, so sole reliance on 96-track drives may limit you somewhat here.

And don't forget that you can also shop for software deals for Kaypro, Osborne, Epson, and other CP/M formats. All you need to do is make sure the program includes a terminal installation program. If it does, it's sure to include the H19 terminal.

I know I've said it before, but I'm still absolutely in love with Write-Hand-Man from TMSI. I've covered it in detail in previous columns, and for the price it's the niftiest, handiest little piece of software you can pop into your system. Again, once you've tried it you'll never do without it. It's also a great show-off piece for those smart-alecks with their MS-DOS machines who think they've got a monopoly on desktop utilities.

Finally, one extra that I don't have yet is a RAM disk. Installing up to one megabyte of RAM that your system treats as a disk drive will really make your computer fly. I've used RAM disks on other machines, and the difference in operation is mind-boggling from the first try. Several vendors are making them for the '89, and prices are really pretty reasonable.

United we stand

So what do you gain from all this?

Once you've gotten a basic "survival system" and opened methods of communication, then distribution of information and software will be much easier.

This column and other "information exchanges" (bulletin boards, newsletters, etc.) can be a central element in the survival of our eight-bit systems. Write to me with your information to share. If you need an answer to a question, chances are somebody has already found it; it's just a matter of getting the answer spread around.

With a little joint effort, we just might be able to stir up some of that kamikaze underground feeling of togetherness that microcomputer users used to share.

The quest for the perfect editor

No sooner had my column recommending VDO as a public domain editor gone to press than I replaced it with another—VDE. VDE adds even more features to VDO and, most importantly to me, adds word wrap. (I talked about VDO in this column in *Sextant* #23, July-August 1986.)

However, before I even got a chance to write my new recommendation, I stumbled across my life-long favorite, PIE, buried in a close-out pile at HUGCON.

Temptation got the best of me, and I'm now running PIE under both HDOS and CP/M. To be honest, it feels like coming home.

Volunteer needed

Some of you have taken me up on my previous offers of copies of the public domain software I mention in my column; if so, you know that my turn-around time leaves a little to be desired. I guess my thirty or so other projects just bog me down a little too much; I'm afraid I'll have to discontinue the practice.

However, if someone else out there would like to take on the responsibility of distribution, drop me a note. I will provide one copy of each public domain program I mention each column to one person, and then he or she can take over the mailing process.

I warn you, even though you'll be working with a supposedly "obsolete" system, your mailbox will fill up awfully fast.

Write!

If you've got something to get off your chest, just drop me a note. If you're using U.S. mail, be sure to include a self-addressed stamped envelope if you'd like a written reply. However, remember that electronic mail gets answered first.

Walter J. Janowski
1505 East Evans Avenue
Valparaiso, IN 46383
GENie Mail: WJJANOWSKI
CompuServe: 72376,1652

Suppliers Mentioned

Analytical Products
213 Teri Sue Lane
Buellton, CA 93427
805/688-0826

Buss: The Independent Newsletter of Heath Co. Computers
716 E Street, S.E.
Washington, DC 20003
202/544-0900

C.D.R. Systems
7210 Clairemont Mesa Blvd.
San Diego, CA 92111
619/560-1272

Echelon, Inc.
101 First St.
Los Altos, CA 94022
415/948-5321

Heath Company
Benton Harbor, MI 49022
Orders: 800/253-0570

Heath/Zenith Users' Group
Hilltop Road
St. Joseph, MI 49085
616/982-3463

H-Scoop
2618 Penn Circle
Sheboygan, WI 53081
414/452-4172

Kres Engineering
P.O. Box 1268
La Cañada, CA 91011
818/957-6322

Magnolia Microsystems
2820 Thorndyke Ave. West
Seattle, WA 98199
206/285-7266

Newline Software
P.O. Box 289
Tiverton, RI 02878
401/624-3322

SigmaSoft and Systems
17000 Dallas Parkway, Suite #207
Dallas, TX 75248
214/380-6187

The Software Toolworks
1 Toolworks Plaza
13553 Ventura Blvd.
Sherman Oaks, CA 91423
Orders: 800/223-8665; in California,
800/228-8665

TMSI (Technical Micro Systems, Inc.)
366 Cloverdale
Ann Arbor, MI 48107
313/994-0784

Workman and Associates
1925 East Mountain St.
Pasadena, CA 91104
818/791-7979

△



Controlled Data Recording Systems Inc.

Quality Products and Support for the Heath/Zenith Community

New for the H/Z89-90 Computer Users: The Super RAM 89 Package

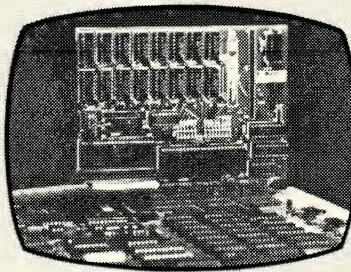
Now get the speed and power of a high capacity Ram Drive System at a reasonable price.

Using the Ram 89 package, standard software shows an immense improvement in speed. Depending on the software being run, programs may execute 10 times faster than when run through standard floppies.

Now get RAM 89

HDOS DRIVER Software Only \$35

RAM 89 DVD supports RAM SY0: and SY1:, added autoload features, plus more!



Ram 89 comes as a two card set that plugs into the left hand (16k expansion) side of the computer. No computer modifications required.

Board 1 has two banks of 256k chips possible for a total of 512k. Either or both banks are able to use 64k chips instead. Board 1 can be used by itself, with board 2 added at a later date.

Board 2 has an additional 512k, plus it has a real time clock capability, and a SASI interface hardware capability. Board 2 piggybacks onto board 1.

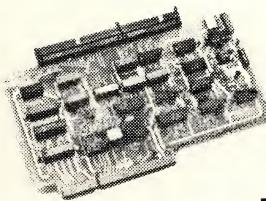
The Ram Drive Software (SRAM) allows one or two logical ram drives. The ram drive(s) can be located starting anywhere from logical A: to O: (standard drives get relocated). SRAM can be set to start at logical A: and warm boot with ram (no floppy disk accesses needed). Ram drive attaches to any of the versions of CP/M 2.2 bios used in the H/Z89-90.

GET A FULL MAGABYTE OF RAM FOR YOUR H/Z89-90

Board 1 includes hardware manual and ram drive software with no ram: \$190.00. Each 256k bank, add \$56.00. Board 2 (must have board 1). With no ram, no clock, no SASI \$90.00. Each 256k bank, add \$56.00. Ask about clock, SASI pricing.

FOR THE H/Z89-90 COMPUTERS

THE ONE CONTROLLER FOR 8" & 5.25" DRIVES THE FDC-880H PRICE: \$395



Includes controller board CP/M boot prom, I/O decoder prom, hardware/software manuals BIOS source listing. HDOS driver now available for \$50.00.

Now be able to run standard 8" Shugart compatible drives and 5.25" drives (including the H37 type) in double and single density, automatically with one controller. The FDC-880H operates with or without the Heath hard sectored controller.

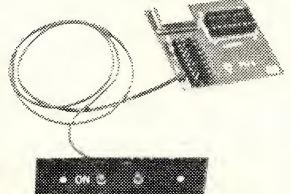
Z100 HARDISK CONTROLLER

The CDR317 is the ULTIMATE IN HARD DISK controllers for the H/Z110-HH/Z120 computers

- o Direct replacement for the Heath/Zenith Z217 hard disk controller package.
- o Able to work with standard ST506/412 type Winchester Drives and the cartridge removable Syquest Drives.
- o Works with all standard Heath/Zenith drivers, operating systems, and Z217 Boot Roms.
- o Is an intelligent SCSI host adaptor for future network expansion.
- o Less expensive than the Z217 controller.
- o Available now through your local Heath/Zenith dealers.

FOR THE Z100 COMPUTERS

THE ORIGINAL Z100 SPEED MODULE. RUN YOUR Z100 PRO- GRAMS FASTER



The ZS100 runs the Z100 CPU 50% faster, (7.5 MHZ) in 8088 mode. The ZS100 installs easily with no soldering. The ZS100 is externally switchable between speed mode and normal. The ZS100 improves the time performance of applications packages with no software modifications needed. The ZS100 is for all Heath/Zenith 100, 110 and 120 series computers.

SUPER DRIVES 85

A powerful floppy drive package for the Z100 or any computer that can use 8" floppy drives.

Two extended technology 5.25" drives in a package with interface cables, case and power supply, that look exactly like standard 8" floppy drives to the computer. Now have **2.4 megabytes of floppy file storage capacity** in one 5.25" package.

ONLY \$495 LIST
Dealer inquiries invited.



For information about these products contact
Controlled Data Recording Systems Inc.
7210 Clairemont Mesa Blvd
San Diego, Ca. 92111 Phone (619) 560-1272 or contact a Heath/Zenith dealer that carries the C D R Systems Inc. line of products
New - C D R Systems Bulletin Board (619) 560-8929

Circle #108 on Reader Service Card

Four Upgrades for the '89

Upgrading your '89 can be a good alternative to buying a new system. Here's a look at four upgrades you might want to make, and how to make them. These upgrades will give you 512 to 1024K of RAM disk space, the ability to interface with a parallel printer, the chance to utilize an additional 1K of HDOS's low memory, and the ability to use 2 or 4 MHz as you need it.

William Clarkson

I built my Heathkit H89 computer in the summer of 1980. An H14 printer and H77 outboard disk drives followed shortly thereafter. They have served me well for six years, including supporting my activities as an independent Heath/Zenith software developer. (Two of my programs are S-BASIC and DISKSORT, both distributed by Sunflower Software. My own company is Comptographics.)

Fortunately for me and my family, I also have a real job! At work, I have a Z161 in my office, a Z100 in the outer office, and Z150 compatibles all over the place. So eventually I would feel constrained by my H89 system, and replace it with a newer computer, right? Right on the first part; wrong on the second!

Oh, I contemplated getting a newer computer. The H/Z158 was very appealing. And if cost was to be the dominant factor, there were the "generic" IBM clones.

Then I looked at the software tradeoff. I listed the major software products I have and use regularly, and what I would have to replace them with (in most cases, the same software products I use at work). It came out something like this:

Editor

Have: PIE, The Software Toolworks (\$29.95). Does everything I need; operates entirely in memory; small enough for me to have a copy on every bootable disk; concise manual.

William Clarkson is a partner in Comptographics, a software development firm. He has over 25 years of experience in software development and management in the aerospace industry.

Would need: P-Edit, SSI Software (\$95). Fine editor, but more functions than I need or can remember; accesses the disk frequently to maintain "spill files"; big (many files); "no sense of humor"; generic (i.e., confusing) manual.

Word processor

Have: TEXT, The Software Toolworks

"How things work together is the responsibility of the Systems Integrator. If you bought a system or subsystem from a single source, then that supplier is responsible for those pieces working together. If you bought components or subsystems from multiple suppliers, then you have assumed the role of Systems Integrator and must expect to perform that function. . . ."

Quick-P User Manual, McGaffey Engineering.

(\$39.95). Does everything I need; one good-sized file; comprehensive manual; not "what you see is what you get"; instead, all print-formatting commands are visible in the text.

Would need: WordPerfect, SSI Software (\$495). "Full-featured" word processor. Same commentary as for P-Edit (it's a companion product); "what you see is almost what you get" mostly; "invisible" commands.

Spreadsheet

Have: MYCALC, The Software Toolworks (\$59.95). Able to do my income tax. (That's the biggest problem I ever want to tackle with a spreadsheet!)

Would need: Lotus 1-2-3, Lotus Development Corp. (\$495). The universal standard.

Telecommunication

Have: REACH, The Software Toolworks (\$19.95). (The first program I bought!) Does everything I need; small enough to have individually configured versions for

each remote system I work with; powerful command language; concise manual. *Would need:* Smartcom II, Hayes (\$149). Does the job but big (many files); seems to take forever to initialize; comes with canned capabilities to talk to every time-sharing service under the sun (most of which I couldn't care less about); command language less capable than REACH; large, poorly organized manual.

C compiler

Have: C/80, The Software Toolworks (\$49.95, plus \$29.95 for optional MATHPAK). Almost the full C programming language; compiles rapidly; produces object programs efficient in both size and speed. *Would need:* Lattice C, Lattice (\$500). The full C programming language; the best regarded and

mature C compiler for the Microsoft Disk Operating System (MS-DOS). But look at the price.

The conclusion was obvious: If I upgraded to an MS-DOS computer, I would have to spend *big bucks* for software, just to get the equivalent of my current capabilities (which, of course, are already paid for!). In fairness, though, I would wind up with more capability.

(As an aside, it appears that parsimonious programming is becoming a lost art with today's powerful microcomputers with large memories. Too bad!)

So, instead of replacing my venerable H89 computer system, I upgraded it to overcome three shortcomings that were particularly constraining: poor print quality from the H14 printer, limited working disk space due to the 100-kilobyte hard-sectorized disks (I never installed the soft-sector controller), and slow speed (compared to MS-DOS computers).

Four changes

The products I installed were:

- The Quick-P parallel printer interface board from McGaffey Engineering (\$69.50, plus \$4 shipping). This board makes it possible to interface the H89 with a modern parallel printer. (I bought an Epson FX-85.) The Quick-P was described in *Buss* #111, and has been advertised in *Sextant*.
- The Super RAM 89 memory-expansion board with CP/M from C.D.R. Systems, Inc. (\$190, plus \$56 for each 256-kilobyte bank of memory chips, plus \$35 for software for the Heath Disk Operating System—HDOS). This board provides 512K of random-access memory (RAM) for use as a RAM disk.

An additional 512K is available with an expander board (\$90, plus \$112 for the memory chips). With the expander board, you can also get a real-time clock (\$45 extra; I have not yet gotten it).

The Super RAM 89 was described in *H-Scoop* #66 and *Buss* #106, #112, and #113, and has been advertised in *Sextant*.

- The Speed Mod from Micronics Technology, which provides software-selectable 2- or 4-megahertz operation (\$24.95 in kit form; \$34.95 assembled). It was described in *Buss* #117.

- The HDOS Low-Memory Kit from Horn Engineering Associates (\$19.95, plus \$1 shipping). This kit consists of two chips to add 1K of low memory with HDOS, and a disk with software that utilizes that memory (including the "software clock" described in the article by Charles E. Horn in the November-December 1985 *Sextant*). This item was just for fun!

To give you the "bottom line" right at the top: all these products *work*, and *work well* (hardware and, in most cases, software also). The manuals are good. The vendors responded promptly to my pre-purchase inquiries, and shipped their products reasonably promptly after receiving my orders.

If that were all, there would be little point in my writing this article. I did have difficulties in the activity known as "hardware/software integration." The dif-

ficulties were mostly overcome, however, and therein lies my story.

While I was coping with some of the difficulties, however, it occurred to me that I might be able to make improvements on the software support for some products. And having written the software, I decided I might as well include it on a disk of utilities that I offer through Comptographics.

The disk is available for \$19.95, and is described in the comments accompanying this article. (See "Comptographics Utilities Disk.") Also, I'll refer to these utilities when I discuss some of the problems I ran into.

I hope my findings will save some time and trouble for others slogging down a similar path.

Quick-P parallel printer board

First, I should note that the Quick-P board is no longer available from McGaffey Engineering. They discontinued production in the fall of 1986. (McGaffey Engineering stated in *Buss* #130 that if anyone is interested in

I have happily thrown away my bowl of spare printhead fuses.

purchasing the commercial rights to manufacture the Quick-P, they would be open to negotiation.)

If you're an inveterate kit builder, however, there may be hope: McGaffey is offering the Quick-P schematics, manual, and printed circuit boards for \$15, and just the schematics and manual for \$5.

To McGaffey's documentation and parts list, I would add that some of you may also require two #4 hex spacers, two #4 lockwashers, and two 4-40 nuts. The documentation and parts supplied with the original Quick-P package assume that the Quick-P board will replace an existing port on the serial input/output (I/O) board (e.g., port 340). But many early H89s (such as mine) were delivered with the 320 port unpopulated. Since one of the first steps in installing the Quick-P is to pull out the port chips, the natural thing for me to do was to install the Quick-P at port 320, and keep my serial port at 340.

The original package, however, provided no hardware for mounting the DB-25 connector on the rear panel. (I guess it is assumed you will use your old hardware.) The "extra" hardware isn't expensive, just hard to find (especially the hex spacers).

Also, I use the REACH modem program from The Software Toolworks. The REACH manual states that, on the serial I/O

board, you must set the interrupt jumpers for your printer and modem ports to priority level 3. So when I installed the Quick-P, I set the port 320 interrupt jumper to 3. When I turned the power on, I not only didn't get the reassuring "beep-beep," I didn't get the H: prompt or any response whatsoever from the computer.

On a hunch, I moved the jumper to the "off" position. That solved the problem, and didn't seem to cost me any capabilities. I found I could still "echo print" when using REACH, or "save" a portion of a session on the LP: "file."

With HDOS, the Quick-P works beautifully with my Epson FX-85 printer and the UD.DVD Universal Printer Driver from SoftShop. That driver allows you to have eight "virtual printers," each individually configured. For each "virtual printer," you can specify an ASCII string of up to 16 characters to be sent to the printer before a print operation. This makes it easy to command the Epson. So you can configure LP0: for standard draft pica, LP1: for elite indented eight spaces, etc. The FX-85's NLQ ("near letter quality") mode is a vast improvement over the H14's tailless p's and q's, and I have happily thrown away my bowl of spare printhead fuses!

I wanted an equivalent capability to initialize the FX-85 printer under CP/M. So, I wrote an enhanced version of the program described in the article entitled "Setting the Attributes of Your Console and Printer" by William Pierpoint in the December 1985 issue of *REMark*.

My program is called SetPrinter (SP.COM). It's available on the utilities disk mentioned above.

An unsolved puzzle

There is, unfortunately, one small fly in the ointment: The Quick-P advertising and documentation implied that all your software will run exactly as it did with a serial printer. Not quite true!

In my initial experimentation with the Quick-P, I noticed that printouts that used to end with a form feed stopped in the middle of a page. This occurred with CP/M, and also with HDOS using Heath's H14 device driver (LPH14.DVD), which McGaffey had recommended. Curiously (and fortunately, for me), the problem did not occur with HDOS and the UD.DVD driver!

After considerable experimentation, I concluded that when a string of characters is sent to the printer via the Quick-P, the last character is not sent out. The missing character (whatever it may be) is subsequently "pushed out" when another string of characters is transmitted, and becomes the first character of that string.

Fortunately, it's easily worked around. It doesn't occur with the UD.DVD driver. My SetPrinter program has ways of taking care of it with CP/M, or when necessary,

Wondering how many issues are left on your subscription?

0.00 30638 23JL 275 -->28

The rightmost number in the top row of your address label indicates the last issue on your subscription. In the sample above, the last issue number is 28, indicating that the subscription ends with the May-June 1987 issue of *Sextant*.

you can just take the printer off-line and hit its form-feed button.

Super RAM 89

Although I use both HDOS and CP/M, I prefer HDOS. So, after installing the Super RAM 89 board, I proceeded to HDOS software installation according to the manual by Al Heigl, the software developer. Let me say at the outset that everything would have probably gone fine if I were running "plain-vanilla" HDOS right off the distribution disks. But who does?

Among the list of possible error reports from the C.D.R program that patches HDOS for the RAM drives is: "Cannot find Patch History Table (should not occur)". Evidently, Al Heigl didn't read "Losing Weight with HDOS 2.0" by Patrick Swayne in the August 1981 *REMark*.

In that article, Swayne described a procedure and provided a program (REDUCE.ASM) for reducing the size of some of the HDOS files by eliminating their patch history tables. I (and many others, undoubtedly) used his technique for gleaned a few more precious sectors on my 100K disks. So, knowing full well what to expect, I ran the Super RAM 89 HDOS patch program, and got the error message that "should not occur"!

Since I had long ago REDUCED *all* my working HDOS disks, the only thing to do was go back to the distribution disks. (That turned out to be a *long* way back. Do you remember setting the terminal so it wouldn't map everything to uppercase?)

Thankfully, my own "patch history table" (on paper) was in good shape, because I had made other patches to HDOS along the way. All of those had to be redone in the process of producing a new working HDOS system disk. Then I could run the Super RAM 89 HDOS patch program. (And finally I could run REDUCE again—because I'm still stuck with those 100K disks!)

Heigl's manual suggests that you might want to make the Super RAM 89 modifications to all your bootable HDOS disks. My advice is: don't do it.

I found there were many times in the process of setting up my Super RAM 89 capability when I needed to go back to those old unmodified bootable disks. Disks are cheap, and besides, you only need *one* Super RAM 89 bootable disk.

So, use a new disk to create a Super RAM 89 bootable disk. (Make a backup too, of course.) Once you boot with that disk and establish your RAM disks, you can just copy the files you want from other disks (bootable or not) to the RAM disks. (More about "disk strategy" later.)

The most effective way to utilize the RAM disks with HDOS is the SWAP option, in which the RAM disks become

SY0: and SY1:. The HDOS system files are copied to RAM SY0: (providing very rapid response to system commands); the physical drives become SX0:, SX1:, etc.

Unfortunately, that brings a few worms out of the woodwork!

I (and many of those other folks who have gone beyond plain-vanilla HDOS) use Super Sysmod2 from SoftShop; this provides short abbreviations for HDOS commands, and some very useful additional capabilities. Unfortunately, the abbreviations can't be used for the renamed SX: drives.

The Super Sysmod2 manual points out, however, that commands preceded by the ordinary slant mark (/) refer to the DK: drives, and commands preceded by the backslash (\) refer to the DY: drives. Since I don't have any of either, I decided to see if I could make the "/" prefix work for the SX: drives.

Fortunately, it was easy. Here's how to do it:

Using a disk-patch utility (such as the Heath Users' Group's DUMP, or Super Zap from The Software Toolworks), examine the third sector of SYSCMD.SYS (sector 002

I got the error message that "should not occur"!

of the file) after it has been modified by Super Sysmod2. About two-thirds of the way down, you will find the letters DK (hex 44 4B). Change them to SX (hex 53 58). The "/" prefix will now reference the SX: drives. (If you prefer to make the change for the DY: drives, you will find the letters DY just a little farther down in the same sector.)

Instead of HDOS's FLAGS utility, I use the PHLAGS program from Sunflower Software's Search and Print package. (It is a lot better in several important ways.) I had a very old version of PHLAGS; when I tried to use it to set a flag on RAM SY1:, it not only refused to recognize my SX: drives, but proceeded to corrupt the disk in my SX1: (nee SY1:) drive. Fortunately, for a nominal price Rick Kerbel of Sunflower Software will provide an upgrade to PHLAGS Version 2.0; the upgrade works fine.

For word processing, I use TEXT from The Software Toolworks. TEXT has a "read file" (.rf) instruction, which incorporates the text from another file into a document it is processing. If you don't specify a drive with the ".rf" instruction, TEXT will look for the file on "... all mounted drives."

As you might have guessed, TEXT won't look on the SX: drives. It does, however, search those wonderfully handy DK: drives, so once again the fix is to change

DK to SX. Right about the middle of the fourth sector (sector 003) of the TEXT.ABS file (version 4.0), you will find DK0:, DK1:, and DK2:. Change the DKS (hex 44 4B) to SX (hex 53 58).

As I previously mentioned, my modem program is The Software Toolworks' REACH. The Options Menu in the HDOS version of REACH allows you to change the disk in SY0:, SY1:, and SY2:. If you exercise one of those options after a SWAP has made the RAM disks SY0: and SY1:, the results are amusing but fortunately harmless: the "disk" in the specified RAM drive is "dismounted" and "mounted" in the twinkling of an eye!

If you change the "disk" in RAM SY0:, all disks (both real and RAM) will be dismounted when you exit REACH, and the system will insist upon being rebooted (again, harmlessly—no files are lost in the process). In short, it is not useful to be able to change the "disk" in a RAM drive!

You might like to change the REACH disk-reset options to SX0:, SX1:, and SX2: (the physical drives). This can be done as follows: In about the middle of the ninth sector (sector 008) of REACH.ABS version 2.0, you will see SY0: (hex 53 59 30 3A). Change it to SX0: (hex 53 58 30 3A). Near the end of the same sector, you will see SY0: and SY1: (hex 53 59 31 3A). Change them to SX0: and SX1:. At the beginning of the next sector (sector 009) is SY2:. Change it to SX2:.

There is one other SY0: in the file, at the end of sector 012. You might be tempted to change it also, but don't. It is the default drive for command files, and should remain SY0:.

HDOS disk strategy

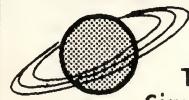
Al Heigl's manual contains a few pages that must be read at least six times slowly for full understanding. One of them is page 16, entitled "INIT and SYSGEN Considerations." For that one, even six readings didn't hack it. I finally settled for his advice in the next-to-last paragraph, which (paraphrased) is: Do your INITs and SYSGENs with one of your old, unmodified bootable disks. That policy allows you to use his "small" RAM-disk device driver RDSMALL.DVD (renamed RD.DVD) without fear. With my 100K physical disks, that's what I did.

Heigl's manual describes extensive automatic capabilities for loading the RAM disks. The problem I saw with that approach is that you can spend a lot of time and do a lot of disk swapping just to get ready to do something fairly simple.

My "disk strategy" for Super RAM 89 with HDOS is quite straightforward: I created *one* bootable Super RAM 89 disk, which executes a SWAP to make the RAM drives the SY: drives, automatically copies system files to RAM SY0:, and then copies a specified fairly small set of my most-used files to SY0:. (The procedure to do

SOLARSIM

by
James Tursa



The Solar System Simulation Program

SOLARSIM is a dynamic 3-D color graphics simulation of our solar system and nearby star systems. It simulates the planets, 250 asteroids and comets (including Halley's), and over 800 stars, nebula and galaxies (you can add more!). View and identify the stars and solar system from any point on Earth or nearly any point in space and at any specified time or date. Great for stellar navigation, tracking comets and planets, and learning the positions and shapes of the constellations. **SOLARSIM** comes with a high-quality 50-page manual. **SOLARSIM** (IBM® version) has both medium-resolution color and high-resolution single color modes.

Only \$29.95 for a limited time!

(Please add \$2 for shipping. Texas residents add sales tax. Overseas orders should add an extra \$5)

Available for the IBM PC/XT/PCjr AT and compatibles, Zenith® Z-100, and the TI® Pro.

System Requirements:

128K RAM & Color/Graphics Adapter (IBM)

192K RAM (Z-100)

256K RAM (TI Pro)

DOS 2 or later (except Z-100)

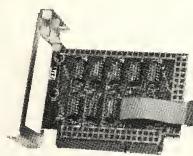
Send check, money order or COD orders to CYGNUS, P.O. Box 57825, Webster, TX 77598, (713) 486-4163

Credit card orders only:
(800) 622-4070 (Nat'l)
(800) 942-7317 (ILL)



Circle #137 on Reader Service Card

HEATH/ZENITH 88, 89, 90 PERIPHERALS

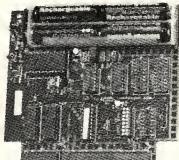


16K RAM EXPANSION CARD

Only \$65.00
Shipping &
Handling \$5.00

REAL TIME CLOCK

Price \$130.00 with
Batteries
Shipping &
Handling \$5.00
\$114.00 without
Batteries



2 PORT SERIAL 3 PORT PARALLEL I/O CARD

Price \$199.00
Second Operating System Driver \$25.00
Shipping & Handling \$10.00



PRICES ARE LESS SHIPPING & TAX
IF RESIDENT OF CALIFORNIA.
MAIL ORDER: 12011 ALCARE STREET,
CERRITOS, CA 90701
(213) 924-6741
TECHNICAL INFO / HELP:
8575 KNOTT AVENUE, SUITE D
BIENA PARK, CA 90620
(714) 952-3930

TERMS & SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE
VISA & MASTERCARD GLADLY ACCEPTED

Circle #157 on Reader Service Card

this is well described in the manual.)

I also created several other *non-bootable* disks—"Utilities," "Text Processing," "Assembler and BASIC," "C/80 Compiler," and "Telecommunication." Depending on what I want to do, I just mount the appropriate disk in, say, SX1:, and use the COPY command to copy some or all of the files on it to RAM SY0:. Somewhat manual I admit, but then I like stick-shift cars, too.

Installing under CP/M

I also installed the Super RAM 89 CP/M software, using the same strategy of *one new* bootable disk plus several non-bootable disks containing files in various categories as described above. My bootable disk is configured to run the automatic command line SUBMIT PROLOGUE on cold boot. (My HDOS heritage is showing!)

The first line of the PROLOGUE.SUB file contains the command ARAM. (This calls the program that automatically establishes the RAM drives; you create it by following clear instructions in the manual.) That's followed by

C:PIP A:=C:*.COM
to copy the .COM files on the disk to RAM A:

If you're running under CP/M, switching to HDOS with Super RAM 89 is straightforward—you just insert your bootable HDOS Super RAM 89 disk into physical drive SY0:, and hit SHIFT-RESET followed by B(oot). Al Heigl's HDOS software will initialize the RAM disks.

Moving from HDOS to CP/M is equally straightforward *the first time*. (No, I'm not kidding!) But say that you initialize the RAM disks with CP/M, then go to HDOS, and then later re-boot from your CP/M Super RAM 89 disk. If you do, the CP/M Super RAM 89 software (ARAM) thinks the RAM disks have already been initialized, and does not alter them. (The clue is that the rest of the SUBMIT file after the ARAM command—the PIP command to copy files to RAM A:—is not executed.)

Since the RAM disks were most recently established as HDOS disks, that leaves them with what looks to CP/M like garbage!

Since the automatic route (ARAM) has problems, you might think of using SRAM—the "manual" way to set up the RAM disk. The CP/M Super RAM 89 manual implies that the SRAM program will always ask you if you want to reinitialize the RAM disks, but it won't. It too says they are already established, and refuses to touch them.

What to do? A crude but effective technique is to turn the computer off and back on again. But that's hard on the CRT. You might think the installation program INSRAM.COM (which you run when you create your bootable Super RAM 89 disk) would work, but it has the same problem as ARAM and SRAM—it thinks the RAM

disks are already initialized.

I think the best approach is to include the Super RAM 89 test program RAMTST.COM on your bootable CP/M Super RAM 89 disk, and invoke it via C:RAMTST after booting. It takes a minute or so to run, but leaves the Super RAM 89 memory in a state that the ARAM program recognizes as uninitialized. So when the RAMTST program finishes, SHIFT-RESET B(oot) will bring up CP/M Super RAM 89.

Bug note

At this writing, there is still a fairly serious bug in the HDOS support software. (Adolph Stumpf described it in a note in *H-Scoop* #76, July 1986.)

The essence of the problem is that you can't safely copy files between two physical SX: drives. Sometimes, errors will show up in copying; sometimes, the directory on the receiving disk will be

Never copy from one physical drive to another.

destroyed, leaving you with a corrupt disk.

C.D.R. has acknowledged the problem, and has been in touch with Al Heigl about it. But I don't believe that a patched version is yet available (as of late fall 1986).

In the meantime, my rule is simple: *Never* copy from one physical drive to another.

The Speed Mod

The Speed Mod hardware from Micronics was easily installed, and worked perfectly. The software was not so easily installed, and didn't work perfectly!

In fact, it didn't work at all with the Super RAM 89. But I made the necessary modifications to the software, developing a few entirely new software components along the way, and now it all works perfectly. Therein lies my story, but I'm getting ahead of myself. . . .

There have been a number of *REMark* articles describing do-it-yourself approaches to modifying the H89 for 4-MHz operation. These include:

1. "A Programmable 4 MHz Modification for the H89," Pat Swayne, *REMark*, November 1982.

2. "4 MHz Update," Pat Swayne, *REMark*, March 1983.

3. "Changing Gears" in Your 4 MHz H89," Peter Shkabara, *REMark*, July 1983.

4. "A Plug-In 4 MHz Modification for the H89A," Gary Wintergerst, *REMark*, October 1983.

5. "Z80 SPEEDFIX for H8/H89," Frederick F. Freeland, Jr., REMark October 1984.

While there is much useful information in these articles, the techniques they describe require soldering, trace cutting, and wire wrapping—very scary even to some Heathkit builders!

An alternative "high-priced" approach is described in Alan K. Gideon's article in the September-October 1984 issue of *Sextant*, "Dual Speed on Your '89." Just buy a ready-made, plug-in unit complete with extensive manual and software. That approach is exemplified by the Kres Dual Speed Module. Described in Gideon's article, the Kres board sells for about \$100.

At one-third of that price, the Micronics Speed Mod represents an attractive middle course. This article will help you fix its software shortcomings.

(The Micronics Speed Mod has also been discussed in Walt Janowski's column in *Sextant*. See "The Eight-Bit World" in issue #24, September-October 1986.)

Simple installation

I had no difficulty with Speed Mod hardware installation. It consists of replacing U502 (the system-clock chip on the CPU board) with a neat little piggy-back board, connecting one jumper, and replacing the Z80 CPU with a faster Z80A (which is included).

When I installed the Super RAM 89, I had relocated the Z80 CPU onto the C.D.R. board, as instructed. So I simply replaced it with the Z80A. That might be a problem with the Kres Dual Speed Module, since evidently the CPU goes on its board. (Don't these vendors talk to each other? Of course not!)

Changing speed—hardware

A brief digression to explain how the Speed Mod works: The H89 operating speed is set to 2 or 4 MHz by a particular bit of the H89's *control port* (362 octal = 0F2 hexadecimal). With HDOS, the control port is referenced by the symbol OP2.CTL, which is defined in the HDOS assembler-support file MTR.ACM. You can only *write* to the control port; you can't read it to determine its value (which is sometimes useful). Therefore, a companion memory location has been designated to store the current control-port value. I will call this the *control byte*. With HDOS, it is .CTL2FL, which is located in memory at address 40066 (split octal), also defined in MTR.ACM. With CP/M, the control byte is found at location 0D hexadecimal.

The Speed Mod uses bit 4 of the control port to set the speed (counting the bits right to left beginning with 0). Bit 4 = 0 specifies 2 MHz; bit 4 = 1 specifies 4 MHz. The Speed Mod instructions and several of the REMark articles explain

```

TITLE  'MEMTEST.ASM -- 4 MHz Memory Test'
STL   'W. K. Clarkson -- 13-Mar-86'
* This program performs a 4 MHz memory test. It sets the H-89 control
* port speed control bit and jumps to the ROM-based memory test program.
* It is based on the program by Patrick Swayne in REMark Issue 34,
* November 1982, p. 26.

```

EXX	EQU	331Q	Z-80 instruction
MTRTYP	EQU	40071A	Monitor type (MTR90, etc.)
OP2.CTL	EQU	362Q	H-89 control port
ROMTEST	EQU	7122A	ROM-based memory test entry point
SPDBIT	EQU	10H	Control port speed control bit
USERFWA	EQU	42200A	Program start
START	ORG	USERFWA	
	LDA	MTRTYP	
	DB	EXX	
	MOV	L,A	
	DB	EXX	
	MVI	A,SPDBIT	
	OUT	OP2.CTL	
	JMP	ROMTEST	
	END	START	

Listing 1. MEMTEST.ASM uses the '89's ROM memory-test routine to see if your system can run at 4 MHz. Ordinarily, running the ROM routine would cause a switch to 2 MHz. MEMTEST, however, jumps to the ROM routine beyond where it would cause the speed switch.

that with CP/M the H89's clock interrupt automatically sends the value in the control byte to the control port every 2 milliseconds.

So with CP/M, if a program wants to update the control port, it only needs to store a new value in the control byte. With HDOS, for reasons that are totally obscured, it is apparently necessary to *both* store the new value in the control byte and output it to the control port!

Another important control-port bit is the *clock* bit, bit 1, which controls the 2-millisecond clock interrupt. We'll see shortly why it's important.

All of the REMark articles on 4-MHz modifications emphasize the importance of testing to determine if the rest of your system (especially the memory) will operate at 4 MHz. It would be nice if you could just invoke the read-only memory (ROM) memory-test routine while operating at 4 MHz; unfortunately, though, it clears the control port when it starts—which immediately switches the Speed Mod to 2 MHz!

Pat Swayne's November 1982 REMark article presents a 4-MHz memory-test program that avoids the problem by initiating 4-MHz operation, then jumping to the ROM memory test beyond where it clears the port. But Swayne's program is in absolute octal; it must be manually entered at the terminal via the monitor's Substitute command, which is a pain. So I wrote an assembly-language equivalent of that program for HDOS, modifying it to set the control-port bit used by the Speed Mod (which is not the same as Swayne's bit!).

My program loads at the standard location, does the necessary initialization, and jumps to the ROM memory test (from

which, of course, it never returns). The program, MEMTEST.ASM, is given in Listing 1. It ran with no complaints (at twice the normal speed), so I concluded that the hardware (mine and the Speed Mod's) was working fine at 4 MHz.

Changing speed—software

So why, you might ask, do we need software?

Two reasons: You *may* want to change speeds, perhaps to run a game at 2 MHz that is just too fast at 4. (Try The Software Toolworks' SNAKE at double speed!) Or you may want to run the occasional application program that refuses to cooperate at the higher speed.

Also, the software that reads and writes disks (the SY.DVD device driver in HDOS; the BIOS in CP/M) *must* switch to 2 MHz before accessing a disk drive. In fact, that's the trick that the Speed Mod uses to eliminate any need for soldering, trace cutting, or other hardware evils. The Speed Mod manual points out: "Since you are waiting on the disk anyway, this does not cause the H89 to function any slower than if it ran at 4 MHz all the time."

Unfortunately, the software provided to implement that rather elegant concept didn't quite do the job.

Speed Mod support programs

The Speed Mod comes with a CP/M disk containing a number of files for both CP/M and HDOS, including some whose purpose still remains a mystery to me. So I will discuss only the files I used, beginning as usual with HDOS. (Fortunately, I had a utility to copy the HDOS files from the CP/M disk.)

It is necessary to modify SYDVD.ASM as supplied on the HDOS distribution disks

```

STL      'Speed Mod Speed Switch'
EJECT
SY1      PUSH H      (H), (L) -> stack
        PUSH PSW    (A), (PSW) -> stack
        LDA  .CTL2FL control byte -> A
        ORI  2      set clock bit = 1
        MOV  H,A    (A) = control byte -> H
        ANI  10H    extract speed control bit
        JZ   SPDSKP1 if bit = 1 (4 MHz):
        MOV  A,H    (H) = control byte -> A
        ANI  0EFH    set speed control bit = 0 (2 MHz)
        STA  .CTL2FL store (A) = 2 MHz control byte
        DI   disable interrupts
        OUT  OP2.CTL (A) = 2 MHz control byte -> control port
        EI   enable interrupts
        CALL MYSY1  (stack) -> A, PSW
                  (H) = prior control byte, (L) <-> (stack)
                  perform disk operation

        XTHL
        PUSH PSW
        MOV  A,H    (H) <-> (stack) = prior control byte
        ANI  10H    extract speed control bit
        JZ   SPDSKP2 if bit = 1 (4 MHz):
        MOV  A,H    (H) = prior control byte -> A
        STA  .CTL2FL store (A) = 4 MHz control byte
        DI   disable interrupts
        OUT  OP2.CTL (A) = 4 MHz control byte -> control port
        EI   enable interrupts
        CALL MYSY1  (stack) -> A, PSW
        POP   H      (stack) -> H, L
        RET
SPDSKP1: POP
POP
RET
return to original calling routine

```

Listing 2. These lines are to be added in the appropriate place in the source-code file for Speed Mod's MTHDOS.DVD. (See text for discussion.) This insertion will ensure that disk accesses do not have the effect of switching the system's speed to 4 MHz.

```

ALLOC: XRA    A      ; ]
STA    UNACNT ; ] standard
INR    A      ; ] BIOS
STA    RSFLAG ; ]

; Speed Mod Speed Switch:

RWOPER: PUSH  PSW    ; (A), (PSW) -> stack
        LDA   CTLPRT ; control byte -> A
        ORI   2      ; set clock bit = 1
        STA   STMHZ  ; (A) = control byte -> STMHZ (temporary)
        ANI   10H    ; extract speed control bit
        JZ   SPDSK1  if bit = 1 (4 MHz):
        LDA   STMHZ  ; (STMHZ) = control byte -> A
        ANI   0EFH    set speed control bit = 0 (2 MHz)
        STA   CTLPRT ; store (A) = 2 MHz control byte
        OUT   H88CTL ; (A) = 2 MHz control byte -> control port
        CALL  RWOPR  ; (stack) -> A, PSW
                  perform disk operation

        PUSH  PSW    ; (A), (PSW) -> stack
        LDA   STMHZ  ; (STMHZ) = prior control byte -> A
        ANI   10H    ; extract speed control bit
        JZ   SPDSK2  if bit = 1 (4 MHz):
        LDA   STMHZ  ; (STMHZ) = prior control byte -> A
        ANI   0EFH    set speed control bit = 0 (2 MHz)
        STA   CTLPRT ; store (A) = 4 MHz control byte
        OUT   H88CTL ; (A) = 4 MHz control byte -> control port
        CALL  RWOPR  ; (stack) -> A, PSW
                  return to original calling routine

STMHZ: DB    0      ; temporary storage for control byte

RWOPR: XRA    A      ; ]
STA    ERFLAG ; ] standard BIOS except
LHLD  DPBX   ; ] label RWOPER changed to RWOPR
MOV   C,M   ; ]


```

Listing 3. This modification of CP/M's BIOS is a replacement for the Speed Mod file SP2.MOD. This is a variation of the author's HDOS code. Adding these lines to the source-code file, though, may make the file too large for some editors. If it is too large, you can use the BREAK utility (included on the Comptographics Utilities Disk) to break up your file into smaller files.

to make it switch to 2 MHz before accessing a disk drive, and then assemble the modified version to produce a new SY.DVD. The HDOS instructions included with my Speed Mod, which were dated 6 Jan 86, explained how to do that. The additional code required was included with the Speed Mod software in the file MTHDOS.DVD.

The Speed Mod HDOS instructions note that "all disk functions pass through the (SYDVD.ASM) code at label SY1." It is not possible for a program to "fall into" that label; the program must be explicitly directed to go there, since the preceding instruction is a RET(urn). So, the label SY1 is changed to MYSY1; the first instruction of the new code is labelled SY1 so all disk functions will now pass through it. The new code (after it does its thing) performs the normal device-driver disk function via CALL MYSY1. The RET following MYSY1, that formerly exited the driver, now returns to the remainder of the new code, which does its wrap-up and then executes a RET.

(This concept of "borrowing the RET(urn)" is a very powerful idea that has general applicability, but it obviously must be employed carefully!)

Some problems

There are several problems with the additional Speed Mod code, however:

1. One more XTEXT statement is needed during assembly—XTEXT.MTR—to include the control-port and control-byte definitions.

2. When the control port and byte are set by the new code, bits other than the ones the Speed Mod should be concerned with are set to 0. (I'll explain why that's bad in a minute.)

3. After completion of a disk operation, the control port and byte are set for 4-MHz operation, regardless of their state beforehand. (So it's impossible to stay at 2 MHz if you access any disks!)

Before I go on to explain what I did to fix those problems, I need to describe another program provided with the Speed Mod. Called SPEED, this program's purpose is to toggle the speed between 2 and 4 MHz. (I bet you were wondering how you ever get to 4 MHz in the first place!)

SPEED is provided in two versions for each operating system—SPEED.ASM and SPEED.ABS for HDOS, and SPEED.C and SPEED.COM for CP/M. Regardless of operating system, the program simply inverts the state of the speed bit (bit 4) of the control byte, leaving the other bits unchanged. It doesn't output to the control port, and it doesn't set the clock bit.

So what? you say—it's okay not to output to the control port with CP/M. (Right on that part!) And why do we need to set the clock bit as long as we leave it unchanged?

Surprisingly, I discovered that, at least

with HDOS, the clock-enable bit as read from the control byte is always 0. So, when you store into that byte and output to the control port, you must always explicitly set the clock bit to 1! If you don't, the 2-millisecond interrupt will be inoperative. A number of essential operating-system routines depend upon it, so lacking it can produce all sorts of strange results.

(Pat Swayne must have known about the problem with outputting to the control port, because his code sets the clock bit. See reference 1 mentioned above.)

So, let's look at what happens when you try to change speeds: Regardless of the H89's state, invoking SPEED requires a disk access; after the disk access, the computer will be running at 4 MHz (problem 3 above); so SPEED will always toggle the speed to 2 MHz (and dutifully report that fact).

Any subsequent disk access (say, to invoke the game you wanted to run at 2 MHz) will switch the computer right back to 4 MHz! I was reminded of those little boxes whose only function, when you turn them on, is to turn themselves off!

Back to the SY: driver: With the Speed Mod changes, it wouldn't work at all with the Super RAM 89. The culprit turned out to be those other control-port bits (problem 2 above).

The Super RAM 89 documentation for HDOS software includes a section entitled "Warnings," which states: "Proper functioning of the Super RAM 89 board is absolutely dependent on the system staying at an Org 0 configuration. . . . Org 0 operation is enabled by including a certain bit in any byte that is sent out the computer's control port. . . . Typically, programs that would write anything to the control port are ones that affect the . . . clock interrupt . . . or 4 MHz speed change."

To correct all the problems I encountered, I replaced "Addition Number 2" in the Speed Mod source-code file for MTHDOS.DVD with my code in Listing 2, adding it *before* the following two lines in SYDVD.ASM:

STL 'Device Processors: Load'
EJECT

Some design standards

The design requirements of my code were based on important lessons I learned many years ago, as follows:

1. If the speed is 2 MHz going in, don't change it. (*"If it ain't broke, don't fix it."*)

2. Always set the clock bit. (*"It won't hurt you to hear this again."*)

3. Don't change any bits other than the Speed Mod bit and the clock bit. (*"Don't mess with things that don't belong to you."*)

4. On exit, set the speed back to what it was going in. (*"Leave things the way*

you found them.")

The Speed Mod documentation points out that Micronics ran into difficulties over item 4 above. The problem seems to have been where to store the previous value of the control byte. I just pushed it onto the stack, figuring that stack integrity would be preserved through the CALL of MYSY1 (nee SYL). Sure enough, my flag was still there!

I wasn't satisfied with SPEED either, not just because of its previously described deficiencies, but because I wanted more than just a simple speed toggle. I also wanted to be able to explicitly set 2- or 4-MHz operation, and determine the speed setting without changing it.

So, I wrote a fancier version of SPEED. This time, I used The Software Toolworks' C/80, with a compile switch to generate object code for either HDOS or CP/M. My version of SPEED is on my Comptographics Utilities Disk.

With my versions of SY.DVD and SPEED, you can switch at will between 2- and 4-MHz operation, and the speed stays as set regardless of disk operations. Going to 4 MHz immediately upon boot-up is simple—just invoke SPEED via a PROLOGUE.SYS file on your boot disk.

I should note that Micronics Technology, the Speed Mod vendor, subsequently sent me updated Speed Mod installation instructions for HDOS dated 31 Mar 86. I didn't try them. (*"If it ain't broke, don't fix it."*) I did observe that they looked more complex than the previous instructions; however, the code to be inserted in SYDVD.ASM appeared to have some of the same problems as the previous code.

In short, I still prefer the 6 Jan 86 HDOS installation instructions and corresponding MTHDOS.DVD file—along with my code in Listing 2 as a replacement for the Micronics Addition Number 2 (and my version of SPEED from the Comptographics Utilities Disk).

Given the new instructions, my advice would be to follow them selectively. First, ignore the new steps 1 through 8b; their only purpose is to produce an SY: driver capable of initializing disks. But the instructions also state that "Disk initialization . . . must still be done at 2 MHz." So why bother?

Add XTEXT.MTR to the set of XTEXT statements in SYDVD.ASM; change the label SY1 to MYSY1; and insert my Listing 2 code at the point described above.

Follow instruction 8.c; ignore instruction 8.d.

Assemble the modified SY: driver (essentially in accordance with instruction 9, but make sure that the necessary XTEXT files are available).

Note that the resulting .DVD file is only four sectors in size! That's perfectly okay, and is a "fringe benefit" of doing it the way I've just described. The standard SY: driver is actually two programs joined

together into one file—the operating program and the initialization program. You can't initialize disks with your "4-MHz system disk"; so, all you need is the *operating* part of the SY: driver. That's what you get when you assemble SYDVD.ASM.

Ignore instruction 10, but follow the remaining instructions to create a "4-MHz system disk."

(Of course, I still recommend that you buy my Comptographics disk so that you can use my version of SPEED!)

Modifying the CP/M BIOS

I use Heath's CP/M version 2.2.04. Although an "autoinstall" program for that version of CP/M comes with the Speed Mod, I prefer the "stick shift" approach, as I said before. Besides, I looked at the code that would be installed. Although it appeared sturdier than the HDOS code, I decided I would rather install a variation of my HDOS code in the CP/M BIOS. (Perhaps a bit of *Not Invented Here*, I admit!)

My BIOS modification is provided in Listing 3. It will serve as an exact replacement for the Speed Mod file SP2.MOD—it is inserted at label RWOPER, and the original RWOPER (which follows) is changed to RWOPR.

If you have Super RAM 89, *don't* install the code from the Micronics SPI.MOD! It stomps on those other control-port bits.

Z-148 EXPANSION CHASIS

7-Slots 135 W Power Supply
Transmitter & Receiver Card, Cable
Openings for floppy or hard disk drives
\$329

Z-100 EasyWIN Hard Disk Drive By UCI

10, 20 or 30 MB
\$695/\$735/\$799

EasyPC by UCI

\$485

U.S. ROBOTICS MODEMS

Courier 1200 \$269
Courier 2400 \$439

FBE Research

ZCLK: On Line Time & Date - \$75
ZMF100A: 768K on Old Motherboard - \$59
ZRAM-205: Z205 Memory Card Upgrade - \$42

TAPE BACK UP SYSTEMS

PC/AT Int. 10 MB/20 MB \$409/\$505
PC/AT Ext. 10 MB/20 MB \$548/\$623

ZENITH PC's Z-148, Z-158, Z-200

IC's V20, 8088, 8087, 64K DRAMs
SEND FOR Z-100 PRODUCT CATALOG

LIGHTEK

P.O. Box 6493 • Kent, OH 44240
(216) 673-5358

Comptographics Utilities Disk

As noted in the body of the article, besides being a user, I am also a software developer. So, when I ran into some of the problems discussed in the article, it seemed only natural to try to write software to make life easier. And since I have my own company, Comptographics, I decided to offer some of them for sale.

In the article, I tried to indicate work-arounds for problems, and Listings 1-3 give code segments that may be helpful, without your having to purchase any additional programs. Nonetheless, some of you may find these utilities convenient in upgrading your '89.

The Comptographics Utilities Disk costs \$19.95. (Specify hard-sectored CP/M or HDOS format, or \$24.95 for both. California residents should include sales tax.) It contains the following programs:

SetPrinter—SP.COM (CP/M only)

SetPrinter is a command-line-driven utility for initializing Epson FX- and LQ-series printers to print a file with the various printing features available (setting characters per inch, boldface, etc.). If you just type the name of the program, SP, SetPrinter will display a list of its valid command-line parameters. If you type the name of the program followed by any number of these parameters, the corresponding printer control sequences

will be sent to printer LST: in the order that you specified them.

SPEED.COM and SPEED.ABS

SPEED is a command-line-driven utility for use with the Micronics Technology Speed Mod. SPEED 2 sets the speed to 2 MHz, SPEED 4 sets it to 4 MHz, SPEED toggles the speed, and SPEED ? reports the current speed. In all cases, the program tells you what it did, and what the current speed is.

BREAK.COM and BREAK.ABS

BREAK is a command-line-driven utility for breaking up large text files so they can be conveniently edited. You can specify the size of the output files, and the drive on which they are to be written. The program can also perform the inverse "join" operation. To minimize the possibility of error, BREAK tells you what it is about to do and asks for confirmation before proceeding.

TABS.COM and TABS.ABS

TABS is a command-line-driven utility for exchanging tabs for spaces, or vice versa, in a text file. You can specify the tab interval. It is sometimes *desirable* to replace spaces with tabs in a file to make it smaller. It is sometimes *necessary* to replace tabs with spaces in a file in order to print it, edit it, or transfer it to another computer.

It's only for coming up at 4 MHz at cold boot, and there's a simpler way to do that. (Read on!)

The need to modify the source of the CP/M BIOS produced an unexpected problem: BIOS.ASM is a huge file, "too fat to fit" into The Software Toolworks' PIE editor!

Since I absolutely refuse to learn or use CP/M's miserable ED, I wrote a C/80 program called BREAK, which breaks up a large text file into smaller files. It can also perform the inverse "join" operation. (BREAK is on my Comptographics Utilities Disk.)

After modifying the BIOS source code, there are a few more hurdles you must leap. To describe the first of these, I can't improve upon the words in the Speed Mod instructions: "Now run MAKEBIOS following the steps in the Heath manual to assemble your new BIOS." (In other words, be prepared for some slow going if you haven't done this before—or even if you have!)

Then FORMAT and SYSGEN a 2/4-MHz bootable disk; remember to set it up for Super RAM 89 (as previously described)

if you have it.

With the modified BIOS and SPEED, CP/M operation is just like HDOS. To come up at 4 MHz at cold boot, you can invoke SPEED from a SUBMIT file run by an automatic command line (as described above in the section on the Super RAM 89). Or you can invoke SPEED directly by the automatic command line if you don't have Super RAM 89 and there are no other functions you want to perform.

HDOS Low-Memory Kit

After pulling out my CPU board more times than I cared to count in the process of installing and checking out Quick-P, Super RAM 89, and the Speed Mod, I was pleased to discover that I could install the two chips comprising the hardware portion of Horn Engineering's HDOS Low-Memory Kit with the CPU board in place. (Charles Horn described his product in "A Software Clock for HDOS," *Sextant* #19, November-December 1985.)

The disk included with the Horn kit

contains several programs that can reside in the 1K of additional memory provided, the most interesting being the software clock, SCLOCK. I had no difficulty following the instructions to assemble a version of that program that puts the clock display in the center of line 25, where it doesn't conflict with the messages of the PIE editor.

So much for the good news. Once again, hardware/software integration produced a couple of bits of bad news: indeed, the presence of the software clock causes an occasional keystroke to be missed, as Horn pointed out in his *Sextant* article. (If there are any words with missing letters in this article, blame them on SCLOCK!)

Moreover, the clock runs slow, and gets further behind with each physical disk access.

That problem was predicted in "Clock Watcher's Delight" by Patrick Swayne in the November 1981 *REMark*. Evidently the "tick counter," from which the software clock gets its timing information, is not updated during disk accesses. Interestingly, this is less of a problem with Super RAM 89 than it might otherwise be, because most disk activity tends to be with the RAM disks.

There is a clock-calibration factor in the SCLOCK program, just as in Swayne's original program. (See line 29 of the program published in the *Sextant* article, line 32 of the SCLOCK program provided on the HDOS Low-Memory Kit disk.) The value of CAL is set to -1-500 (in two's complement notation), so that 500 ticks will equal one second. Swayne points out that lower values (e.g., -1-495) will make the clock run faster.

Horn's programs can serve as prototypes for writing your own low-memory resident programs. There is even a brief section in his manual on that subject. So if you don't like the clock, find something better to do with that 1K!

I thought a great candidate would be a "skinned down" version of the public domain Hard Copy Device Driver HD.DVD by Walter Rison, Burton Hulland, and Dennis Persinger. So I sent it to Mr. Horn; in a couple of weeks back it came, transmogrified into a low-memory resident version! I feel sure he will make it available, perhaps as part of his HDOS Low-Memory Kit.

In summary

The Quick-P parallel printer board, the Super RAM 89 memory expansion board, the Speed Mod, and the HDOS Low-Memory Kit are all excellent products that I am confident will extend the useful life of my H89 system a few more years.

My anecdotes are not intended to reflect criticism of these products, but only to entertain and, I hope, inform. After all, in the "real world" of large-scale software development, hardware/software integra-

tion is always a bear—why should it be any different with our little desktop computers?

Suppliers Mentioned

C.D.R. Systems, Inc.
7210 Clairemont Mesa Blvd.
San Diego, CA 92111
619/560-1272

Comptographics
1612 Via Arriba
Palos Verdes Estates, CA 90274
213/378-5028

Hayes Microcomputer Products, Inc.
5923 Peachtree Industrial Blvd.
Norcross, GA 30092
404/441-1617

Horn Engineering Associates
1714 Patricia Lane
Garland, TX 75042
214/278-3870

Kres Engineering
P.O. Box 1268
La Cañada, CA 91011
818/957-6322

Lattice, Inc.
P.O. Box 3072
Glen Ellyn, IL 60138
312/858-7950

Lotus Development Corp.
161 First Street
Cambridge, MA 02142
617/492-7171

McGaffey Engineering
1556 Halford Avenue, Suite 169
Santa Clara, CA 95051
408/247-6302

Micronics Technology
449 Barbados Way
Niceville, FL 32578
904/897-4257

SoftShop
191 Leland Farm Road
Ashland, MA 01721
617/443-9693 (6:00-8:00 p.m.)

The Software Toolworks
1 Toolworks Plaza
13557 Ventura Blvd.
Sherman Oaks, CA 91423
Orders: 800/223-8665; in California,
800/228-8665

SSI Software
288 West Center Street
Orem, UT 84057
801/227-4020

Sunflower Software, Inc.
13915 Midland Drive
Shawnee, KS 66216
913/631-1333

ZDRAFT FOR Z100

New Features:

- Print-to-file lets you print your drawings on any MSDOS system, OR create drawings for WatchWordTM to include in text documents.
- Read text files into drawings.
- Read plot data from files.
- Sideways printing and split printing.

ZDRAFT is a graphics editor that lets you use your dot matrix printer to make high-quality illustrations for articles and reports. Unlike others, ZDRAFT can use all the memory in your Z100 to make drawings much larger than the screen.

This ad is a single drawing made with ZDRAFT and some of the fonts which come with it. The drawing was printed on a Tally MT180L and photographed exactly as printed with no paste-up (reduced 2.3:1).

ZDRAFT is \$139.00. Demo disk and manual are \$25.00, refundable with purchase. Info and samples are free.

ETTS, Incorporated, P.O. Box 955, Renton, WA 98057-0955
(206) 226-3916

WATCHWORD IS A TRADEMARK OF S&K TECHNOLOGY, INC

Circle #148 on Reader Service Card

ENJOY YOUR CP/M COMPUTER AGAIN

CompuMagic's put 20 programs into one super CP/M Utility Package. No fancy menus; No flashy graphics; Just flexible, fast, hard working programs that do what you want them to do.

Utility programs aren't sexy, but these programs have a way of making even dull tasks like file copying, renaming, sorting, and even erasing so much easier that it's almost exciting. Directory listings, turnkey program starters, screen capture and more. The programs are described in detail in the more than 50 pages of documentation. Order it now and start enjoying your CP/M computer again. Only **\$45**.

30-DAY MONEY-BACK GUARANTEE. All CompuMagic products are fully functional and fully supported, and come with a full, 30-day, money-back guarantee.

ORDER NOW: Send Check or MO (Personal checks accepted) or call with your MC, Visa, or COD order. NO charge for UPS ground shipping. Add \$2 for UPS COD and \$5 for foreign orders. (MD Res. add 5% tax.) Available in 5" Hard Sector and 5" and 8" Soft Sector Formats. Be sure to specify what you need.

CompuMagic, Inc.TM

P.O. Box 437
Severn, MD 21144
(301) 969-8068

Circle #156 on Reader Service Card

Δ

BOOKS FROM SEXTANT PUBLISHING COMPANY

Heath/Zenith Users:

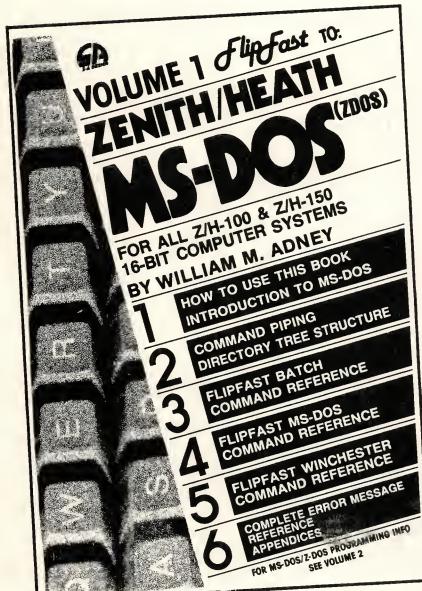
Now you can get the books you need to make your time on the computer more productive—and at drastically reduced prices!

For a limited time, Sextant Publishing Company has lowered prices on the popular *FlipFast Guides* to Zenith/Heath MS-DOS/Z-DOS and Zenith/Heath CP/M-80/85. As an added bonus, we'll send you a FREE copy of *How to Use: Zenith/Heath Computers*, when you buy two or more of the *FlipFast Guides*!

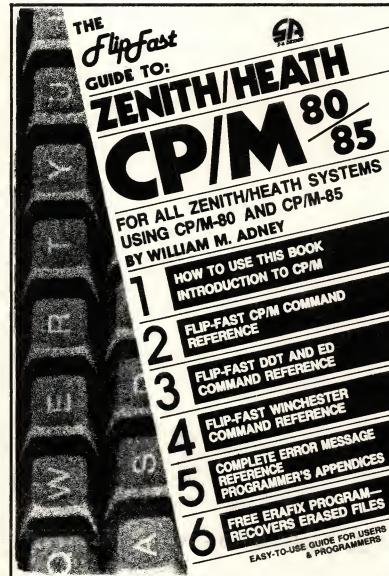
You can buy any combination of the *FlipFast Guides* to get your FREE copy of the *How to Use: Zenith/Heath Computers* book—two MS-DOS books, two CP/M books, or one of each. You just have to buy two books—at **their new low prices**—to get one book FREE!

ORDER YOUR BOOKS TODAY!

Use the card attached above to order your *FlipFast Guides* and your FREE copy of *How to Use: Zenith/Heath Computers!*



Published at \$19.95
Now only \$12.95



**Published at \$12.95
Now only \$7.95**

**Published at \$19.95
Now FREE with purchase
of two FlipFast Guides**

I want to get the most out of my operating system!

Send me

Zenith/Heath MS-DOS/Z-DOS

_____ copies @ \$12.95 each = \$ _____

Zenith/Heath CP/M-80/85

_____ copies @ \$7.95 each = \$ _____

+ \$2.50 shipping charge
per order in U.S./Canada = \$ _____

or

+ \$4.00 shipping charge
per book sent overseas = \$ _____

Total = \$ _____

Name _____

Address _____

Payment enclosed (checks must be in U.S. dollars payable on a U.S. bank)

Charge my: VISA MasterCard

Card # _____

Expires _____

I'm buying two FlipFast Guides—send me my FREE copy of the book, *How to Use: Zenith/Heath Computers*. (For books sent overseas, please add \$4.00 shipping charge to your payment.)

Mail this card today to:
Sextant Publishing Company
716 E Street, S.E.
Washington, DC 20003

26-M/C

I want to get the most out of my operating system!

Send me

Zenith/Heath MS-DOS/Z-DOS

_____ copies @ \$12.95 each = \$ _____

Zenith/Heath CP/M-80/85

_____ copies @ \$7.95 each = \$ _____

+ \$2.50 shipping charge
per order in U.S./Canada = \$ _____

or

+ \$4.00 shipping charge
per book sent overseas = \$ _____

Total = \$ _____

Name _____

Address _____

Payment enclosed (checks must be in U.S. dollars payable on a U.S. bank)

Charge my: VISA MasterCard

Card # _____

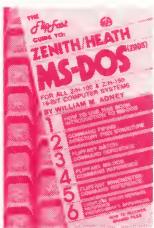
Expires _____

I'm buying two FlipFast Guides—send me my FREE copy of the book, *How to Use: Zenith/Heath Computers*. (For books sent overseas, please add \$4.00 shipping charge to your payment.)

Mail this card today to:
Sextant Publishing Company
716 E Street, S.E.
Washington, DC 20003

26-M/C

Send me the *FlipFast Guide*
to my Heath/Zenith operating system:



**Zenith/Heath
MS-DOS/Z-DOS**

256 pages



**Zenith/Heath
CP/M-80/85**

190 pages

Each guide features

- Easy-to-access information for Heath/Zenith users
- Specific information for each operating command or subcommand
- Easy-to-use listings and tables
- Error message listings

To order your copies, fill out the reverse side of this card today and mail it to
Sextant Publishing Co.
716 E Street, S.E.
Washington, DC 20003

Send me the *FlipFast Guide*
to my Heath/Zenith operating system:



**Zenith/Heath
MS-DOS/Z-DOS**

256 pages



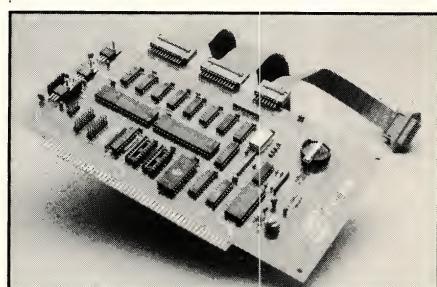
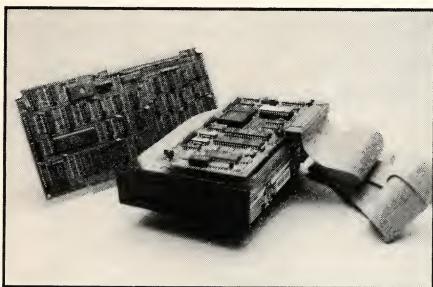
**Zenith/Heath
CP/M-80/85**

Each guide features

- Easy-to-access information for Heath/Zenith users
- Specific information for each operating command or subcommand
- Easy-to-use listings and tables
- Error message listings

To order your copies, fill out the reverse side of this card today and mail it to
Sextant Publishing Co.
716 E Street, S.E.
Washington, DC 20003

UCI Mini Catalog:



EasyWIN

Most economical hard disk system available for the Z-100. It replaces the Z-217 hard disk system and works just like it. You can prep it, part it, and boot from it.

Features:

- Direct replacement of Zenith Z-217 Winchester disk.
- Controls 2 physical drives.
- Supports auto boot, all Z-100 formats, and Zenith diagnostics.
- Operates in any Z-100 with or without EasyPC.
- Future support of other hard disk controllers, removable cartridge drives, etc.
- Operates at 5 or 8 MHz.

Includes:

- Intelligent controller card to adapt IBM PC compatible hard disk controller to the Z-100.
- Western Digital PC hard disk controller card.
- Seagate 10/20/30 MB hard disk drive (ST215/225/238).
- Drive cables.
- Fully documented User's Manual.

Z-148 Expansion Chassis

Finally a way for Z-148 users to take advantage of the growing world of IBM PC hardware including EGA, VGA, modems, hard disk drives, multifunction cards, etc.

Features:

- 8 available slots: 6 full PC slots and 2 half-size slots.
- 135 watt switching power supply with built-in fan.
- UL and FCC approved chassis and power supply.
- 5 or 8 MHz no wait state.
- Standard internal mounting brackets for 2 full-height or 4 half-height drives.
- Easy access, flip top metal case.

Includes:

- Z-148 adapter.
- 5 ft. shielded cable.
- Fully documented Installation Manual.
- Dimensions: 17.5" D x 19.5" W x 5.5" H.
- Weight: 25 lbs.
- Color: Ivory.

I/O Plus

The least expensive way to obtain respectable but limited PC compatibility in a Z-100. Will run many, but not all, popular PC programs. Some copy protected programs may present problems unless patched.

Features:

- ZPC™ program and hardware circuit.
- ZPC program on EPROM (optional).
- One game port (2 joy sticks).
- Up to 115k baud rate.
- 2 IBM 8250 serial ports.
- Clock calendar with battery back-up.
- Fully documented User's Manual.

ZPC is a trademark of Heath Users Group. IBM & IBM PC are trademarks of IBM Corporation. Z-100, Z-148 & Z-217 are trademarks of Zenith Data Systems.

Call now for your copy of the Fall '86 Edition of the UCI Product Catalog.

800-UCI-COMPUTER



CORPORATION

948 Cherry Street, Kent, Ohio 44240
(216) 673-5155 / 800-UCI-COMPUTER

C Notes

Joseph Katz

Microsoft, Si!

...the price

...interlanguage linking

...Xenix compatibility

...the debugger

Screen generators for C

MAJIC's Pro-C

Raney's Screen Editor

Dan Bricklin's Demo Program

C-scape

Zview

Microsoft, Si!

In the few weeks since it arrived, I've found Version 4 of Microsoft's C compiler package extremely appealing. That appeal does not derive from anything dramatic in the compiler part of the package. So far as I can tell, the only really new feature of both the compiler and its libraries is the addition of two new memory models: compact (64 kilobytes of code, 1 megabyte of data) and huge (1M each of code and data, and the potential for arrays larger than 64K). I haven't tried using the compact model yet, and the huge model seems inapplicable to the kinds of programs I usually write.

Everything else in the compiler and libraries strikes me more as repairs and incremental improvements of Version 3 than as the kind of giant leap forward I expect in a major new version. For example, compilation errors are handled better now than before, and executable programs are faster and smaller. But since it was awful of Version 3 to spew an interminable list of error messages when only the first might be significant, I think of Version 4's more sensible approach (aborting after the first screenful of messages) as just a bug fix.

As for better optimization of code by Version 4, it's there and noticeable but not really important to me: Version 3 already wrought small and fast enough for the kind of programming I mostly do.

Version 4 gets its appeal by first stabilizing the shaky points of its predecessor, and then building a new stage upwards. I think I have a pretty good idea of where some things are going, and I like what I think I see.

...the price

Let's get the one *resistible* feature on the table first: \$450. That's the list price for this compiler package. It's designed to be a production package for professional programmers and is priced accordingly.

Nevertheless, there are a couple of ways, not including larceny, to get around the \$450. Many reputable, full-service dealers discount Microsoft C substantially. The Programmer's Shop (800/421-8006) is advertising it now for \$319, and Programmer's Connection (800/336-1166) has it for \$299. I've dealt with the former a few times and am quite taken with their enlightened way of doing business. Once they allowed me to return for a full cash refund some tools that proved wrong for my purposes. I haven't done business with the latter company

yet, but I've heard good reports about them. From either source, you're looking at a purchase price around \$300 instead of one around \$450.

Or, if you bought Version 3 of Microsoft C after June 1, 1986, you can get Version 4 free as an upgrade directly from Microsoft.

If you're serious about C programming—even if you're not a professional—you'll want to look at the new Microsoft C. Everything other than the price of this package strikes me as irresistible.

...interlanguage linking

One of those irresistible features is what Microsoft calls "interlanguage linking," by which they mean the ability to link modules written in other languages into your C program. It was introduced in Version 3 and is retained in Version 4. Now, when you move to C, there's no ineluctable need to discard the investment you might have made in Pascal or FORTRAN libraries: you can link those routines into your C programs. You also can link your C routines into programs written in those languages.

There are catches, of course. Chief amongst those catches are that this feature extends only to FORTRAN and Pascal (in addition to assembler), that the FORTRAN and Pascal must be Microsoft's packages, that you'll need Version 3.3 or higher of Microsoft FORTRAN or Microsoft Pascal, that you are restricted to C's large-memory model, and that you must work with more care than even prudent porcupines exercise when mating.

For example, you must be careful to juggle things so you avoid conflicts in naming and calling conventions in each of the languages: you must observe C's conventions and those of the other language, taking care to avoid any clashes. Make sure, therefore, that no identifier is longer than six characters and that none of those characters is uppercase. Otherwise, you risk having apparently unique identifiers in your C program truncated or uppercased into names already used in the libraries. The result will be either many error messages or, worse, a program that might take off without you in the direction of Tijuana. In short, you're going to have to think simultaneously in C and the other languages to take advantage of interlanguage linking.

What I see here, though, is not merely a set of nuisances. Behind them is evidence of a vision I wholeheartedly ad-

mire, and for which Microsoft deserves applause it seems not yet to have received. That vision is of a potential future in which our many distinct languages remain as dialects from which we can choose according to our needs and inclinations. The vision is directed horizontally—across different operating systems—and vertically—up and down different languages on the same operating system. I like it. I like it so much that I more-or-less gladly bear the burdens of dealing with these first steps towards that future.

...Xenix compatibility

Another irresistible feature of Version 4 carried over from Version 3 is the way it complements Microsoft's C compiler on Xenix—at least on the versions from SCO, the Santa Cruz Operation.

Xenix is Microsoft's implementation of Unix System V for microcomputers. Here are some simplifications, but like many simplifications these contain a great deal of truth: C is the language for Unix, Unix is the operating system that is C's native environment, and Xenix is about as close as you can get to "standard" Unix on a microcomputer. I've been seduced by Xenix, specifically by SCO Xenix, which I have installed on one of my '241's hard disks. I'm still learning Xenix and still experimenting with writing programs for it. Writing programs is much easier with Microsoft's MS-DOS C compiler package because it has been designed as the complement of the C compiler package on Xenix.

Microsoft's MS-DOS and Xenix C compilers have a core library of common functions. If you're careful to use only them, a program can be compiled on one system and used on the other. As with interlanguage linking, there are some catches in writing portable programs. These catches are so picky that they resist my attempts to find any brief examples; still, they're reasonable and not onerous. They are laid out in Appendix G of the compiler's User's Guide, one of three hefty volumes supplied as documentation for Version 4 of Microsoft's C compiler.

Curiously, I find little trouble doing the juggling needed to write programs that will work on both Xenix and MS-DOS. Most of my stuff isn't fancy enough to create problems, I guess, because they're usually written for me or my wife to handle situations we understand. In those programs, I don't have to worry about nice user interfaces or dazzling graphics. The more you do have to contend with such requirements, the less portable your code will be.

Another reason I don't find it much of a problem to write programs usable on either operating system is that the instructions for the MS-DOS compiler reinforce those in the two volumes of documentation for the Xenix Develop-

ment System. There's much to be said in favor of repetition as a learning device, and even more to be said in behalf of working within situations in which basic rules are exactly the same.

...the debugger

What I like most about Version 4 of Microsoft C, though, is its debugger, CodeView. It's one of those gadgets you want as soon as you see it. It's the best software-only debugger I've yet seen. To indicate why that's so, I can't beat the statement in the "Overview" to CodeView in the Microsoft CodeView and C Language Reference Manual (another volume in the compiler documentation): *The CodeView debugger can display and execute program code, control program flow, and examine or change values in memory. Its window interface makes debugging easy. You can view your*

What CodeView lets you do is watch how your program does what it does, even when it's not what you intended it to do.

source code in one window, commands and responses in another, registers and flags in a third, and the values of variables or expressions in a fourth. You can examine the values of global or local variables, either by themselves or combined with other variables in expressions.

Discard DEBUG that comes with MS-DOS. Disregard SYMDEB that comes with Microsoft's macro assembler. CodeView subsumes everything those two debuggers have offered, then goes far beyond them and any other debugger I've used.

You have to do some planning—nothing burdensome—to use CodeView. Since it's a source-level debugger, which will show you the relationship between your code and its results, you want a readable source display. So you ought to put each statement on a separate line (which I think is a good idea anyway) instead of cramming statements together on a line (which is perfectly legal in C). Then remember that two functions of the preprocessing stage in any C compiler are to expand macros and to introduce #include files. Those jobs are finished by the time CodeView gets your program.

CodeView, however, cannot have more than one file open at a time. If you need to debug complex macros or any other stuff from #include files, therefore, you have to circumvent the preprocessor. In that

case, you should work the long way. Make certain the instructions do what you want before you turn them into macroinstructions: debug the program with source statements such as `((x) * (x))`; then, when you're sure everything works properly, extract your definitions into macros such as `#define square(x) ((x) * (x))`.

A slightly different technique applies for #includes: since one of C's glories is its encouragement of modular construction, debug each module before you debug the program built of them. It's bottoms-up debugging, I suppose. Certainly it makes good sense to make sure that each part works before you try to see if the whole does. You also need to compile your code with one of two possible switches to make the object code contain the symbolic information used by CodeView to find its way around your program. Then you link the .OBJ file with the /CO (for CodeView) switch so the symbolic information in the executable program encompasses library functions. That's it.

With CodeView (including its file of help messages), your executable program (e.g., MATTHEW.EXE), and its source code (MATTHEW.C) all in the same directory, you run CodeView followed by the name of your program as its argument: CV MATTHEW. If your program in turn takes its own arguments, you add them as you would when running the program from the command line—except, of course, for the prefix that runs CodeView:

CV MATTHEW EAT COOKIES.

There are several switches you can give CodeView to modify its normal behavior. One of them, /B, will be used by most owners of Heath/Zenith IBM-compatibles because those machines come standard with a Color Graphics Adapter (CGA) board and a composite monochrome monitor. /B forces CodeView to the two-color display necessary for such equipment; otherwise, CodeView detects the CGA board and assumes there's a color monitor attached to it.

Another switch, /M, tells CodeView to ignore a non-Microsoft mouse on the system. That way, the debugger will bypass any conflicts with the alien mouse, but you can still use the mouse if your program needs it. (In practice, either CodeView is less fussy or the aliens are much better at imitating than one might expect: Logitech's Logimouse C7 and SummaGraphics' SummaMouse both work just fine with CodeView. It's as I've often said: The dangers of aliens are often exaggerated.)

What CodeView lets you do is watch how your program does what it does, even when it's not what you intended it to do. When it isn't what you intended, you can trace forwards or backwards to the trouble spots, and set breakpoints towards which the program will move when you need to zero in on the bug's nest. In short, CodeView does every-

thing that it seems possible for a debugger program to do. You can do instant replays of sections just passed, and you can toggle quickly among your C source code, the assembly language instructions into which it translates, and the normal display the user would see. All these are just the high spots of CodeView.

Nobody likes a tease, but those high spots are all I can give you here: the excellent CodeView documentation, with an equally excellent table of contents and index, spans 245 excellent pages.

Unfortunately, CodeView's revelations are not always flattering. A few years ago I wrote a tool to extract all the unique words in a text file, sort them into alphabetical order, and write the sorted words to a second text file. It's kind of a scholar's jackknife, useful in doing stylistic analyses and in building specialized lexicons. In a burst of lyricism I dubbed it WORDS. The default in WORDS preserves original cases, which makes it treat "red," "Red," and "RED" as unique words. That's the treatment most often appropriate to beginning some kinds of analyses of an author's usage. Computer dictionaries, however, function most effectively with words in either uppercase or lowercase. WORDS, therefore, has switches to force the case as required. I'd honed and polished the gadget periodically for years to get it to the point at which it runs like wildfire. Or so I thought.

At the start of my most recent polishing session—really an indulgence in programmer's narcissism—I decided to run WORDS through CodeView. I'm still nursing my wounds: WORDS, it turns out, periodically checks to see if there was a command-line argument to force the letter case. Those checks continue throughout the program's execution, long after the matter has ceased to be of any interest whatsoever. It costs a bunch of wasted time. I know now I did something stupid. I know now where I did it. And as soon as I saw it I knew how to fix it. I will fix it after the hurt goes away.

You won't really understand my appreciation of CodeView until you see it. You must see it. Fortunately, Microsoft makes

seeing it easy. There's a Learning Microsoft CodeView diskette included with the compiler, and anyone can copy it legally. Many dealers will make a copy for you if you bring along a blank diskette. Programmer's Connection offers to supply one for \$1. If you subscribe to CompuServe you can download the demo from the Microsoft Special Interest Group (GO MSOFT and switch to XA1), but be prepared for a big download and big connect charges. You'd do better to scrounge a copy of the demo disk from a dealer.

Microsoft is mighty proud of CodeView, and justifiably so. In fact, I wouldn't blame Microsoft for being mighty proud of Version 4 in its entirety. It's very appealing.

Screen generators for C

I loathe writing code for fancy screens. That's why I don't do them for programs

These handy-dandy tools let you draw your screen, then automatically generate the C source code.

designed to be used by me alone. It's also why I haven't done fancy stuff for things I give other people. I've noticed, though, that most people—especially those who don't program—usually judge a program's worth by its glitz. Let them choose between two programs that do exactly the same thing in almost exactly the same way, and every single one of those people will pick the one that sends out fireworks and plays "The Stars and Stripes Forever" instead of the one that works efficiently without fanfare. It last happened to me a while ago when I found that some colleagues, who once had been thankful for copies, discarded a useful little program of mine in favor of a blatant

beast laden with baubles, bangles, beads, and a "for sale" sign. "Don't you wish you could write a program worth money?" commented a fickle friend after mailing his check.

Since then I've been studying every screen generator I could find. These handy-dandy tools let you draw your screen, then automatically generate the C source code that will reproduce it whenever the program is run. A screen generator, therefore, consists of at least two components.

First is a graphics editor with which you create the screen you want. The ideal, I think, is something like a word processing program with graphics capabilities. You type your text, lay out rules and borders around it, set foreground and background colors for the entire screen and for individual parts of it, and tinker with everything to your heart's content. I think a good graphics editor lets you do as much as possible with as little effort as possible.

There's a hierarchy of capabilities. Take borders, for example. At the very least you should be able to lay down borders by marking the two diagonal corners of the area they should enclose. You want to be able to do the usual single- and double-rule frames, of course, but it's better if you also can select any ASCII or "extended ASCII" character as the border's building block. Then you want to be able to move the border from its original location to another part of the screen so that the more fastidious of you can get things just right, and the less fastidious of us can repair our mistakes. And then you want the ability to change the border character without having to go through the whole business all over again. Extrapolate those capabilities to every other aspect of screen display and you have what I consider the essentials of a good graphics editor in a screen generator.

The second required component is the code generator. Because its product is a file you incorporate or #include in the source code for your program, the code generator has to be matched to your compiler package. Otherwise, the generated

LEATHERETTE DUST COVERS

High quality, hand crafted, custom fitted leatherette dust covers are available for the following listed Heath/Zenith computers and accessories. If your H/Z equipment is not listed write for a price quotation. Custom dust covers can be manufactured if you will provide a diagram with dimensions (actual cost will depend on the size of the cover and will include a \$5 design fee).

H8, H14, H17, MX-80, ZVM-122/135	\$12.95
H/Z 100 all in one, H/Z 19/89/90	\$15.95
Diablo 630 w/tractor, H/Z-29	\$16.95
H/Z 25/125	\$18.95
H/Z-100 Low-Profile color/mono *	\$18.95

* Specify monitor type (i.e. ZVM-135)



ACCESSORIES

	RETAIL	SALE
Microfazer II 128K print buffer	\$445.00	\$349.00
Backup power supply 400 Watt	\$699.00	\$499.00
Printer ribbons black/colors	please call/write	
Computer furniture	please call/write	
Printer, monitor, etc cables	please call/write	
Hard disk tape backup (20MB)	\$1,295.00	\$895.00
CENTRONIC Laser Printer	\$2,395.00	\$1,895.00

To order send a check or money order for the appropriate amount (no CODs or charges) to Kandueazy Computer Software Services (KCSS), 6218 Blossom Lane, Alexandria, VA 22310. Please add \$2.50 for shipping and handling. Virginia residents add 4% sales tax. FREE CATALOGUE

SPECIAL: DYSAN DSDD diskettes (bulk pack) only \$.79 each in any quantity.

Circle #119 on Reader Service Card

```

#include <color.h>

ask()
{
  cls(); gotoxy(3,0); cccputs("Do you want to join DENSA (Y/n)?",LGRN_F+BLU_B);
  cccputs(" " ,WHT_F+BLK_B);
  gotoxy(3,33); scanf(yesno);
}

```

Listing 1. Dennis Raney's Screen Editor created the above code from a menu drawn on the computer screen. Most of the function calls are from Bob Pritchett's C Spot Run function library for Version 3 or 4 of Microsoft C.

code will use alien function calls. For example, if the function itself is available in a library as `crt.cls()` (as in Computer Innovations' C86), there will be big problems if the generated code calls it as `clscr()` (as in Ecosoft's Eco-C88). Look beyond the vendor's name for the compiler and make sure the code generator also matches the compiler's major version number: Version 2 of Microsoft's C compiler is an entirely different compiler from Versions 3 and 4.

All the screen generators for C I've seen so far are ambitious attempts at producing input screens for data processing programs. You design the screen so that input and display-only fields are marked with codes stipulated by the screen generator. The usual way is to mark each character in each field with codes. Sometimes the codes are for extended ASCII characters (above 0xFF); sometimes they're for colors or other attributes reserved for such purposes. The generated screen becomes the front end for code you write to manipulate data input through it.

Many screen generators provide sophisticated input/output (I/O) routines that greatly aid your efforts. Some screen generators leave the job entirely up to you. If your goal is to write a data base handler, you'll likely want a separate I/O management package such as SoftFocus's ISAM Driver and BTree Library, or SoftCraft's Btrieve, or even Raima's extremely powerful db-Vista. They'll help you handle tasks such as sorting the data base and retrieving a record in it. Before you get caught in an expensive weir, though, you ought to see if a packaged data base manager such as Ashton-Tate's dBASE III Plus, Microrim's R:base 5000, or Cosmos's Revelation will do what you need for the same money or even less.

Most screen generators add a considerable number of functions to the compiler packages they supplement. Ecosoft's Eco-C88 and Computer Innovations' C86 are among the few packages that include screen-handling functions. Other compilers can benefit greatly from drawing on the functions in libraries supplied with most good screen generators—if the generator's documentation adequately

explains how the functions work. Alas, many don't.

MAJIC'S Pro-C

Perhaps the worst screen generator I've seen from the standpoint of documentation is Version 1.0 of Pro-C by MAJIC Software. I don't know how good it is in other ways because I have a hard time getting it to do anything except manipulate the sample program thrown in. Even that's rough going. I think the trouble resides in the thing masquerading as a manual, but the problems may be even more widespread. I don't know because I won't devote the large amount of time I would need to figure things out. This package seems intended for people much smarter than I.

Pro-C is billed as "The Professionals [sic] C-Code Generator," a solecism suggesting that no one involved with Pro-C has much regard for written communication. The suggestion turns out to be true.

"Believe it or not, you probably will not need this manual after today!" exclaims the opening of its first chapter. Well, believe it or not, you probably will find the manual almost completely worthless any day. It gives no useful overview of Pro-C, no useful explanation of how its pieces fit together, and no summary of commands. It's neither a manual nor documentation.

Most of the manual is sort of a tutorial on building a sample mailing list manager for the "Pro-C Leisure Club" with interspersed comments on how—but not often why—to do it with Pro-C. It's a recipe for concocting a particular stew.

After you've worked your way through that discursive tutorial, you just might be able to rebuild that same data base management system. I don't think you'll learn enough about Pro-C to do anything else: at least I didn't.

You might well be able to learn how to use Pro-C from its extensive help screens ("1/2 megabyte of context-sensitive online help," the publicity release says), but you'll have to be much sturdier as well as smarter than I: I eventually lost all patience with Pro-C.

If you can muster the brains and pa-

tience to deal with Pro-C, you'll also have to muster the wherewithal for either SoftCraft's Btrieve or SoftFocus's ISAM Driver and BTree Library: Pro-C doesn't have its own I/O engine.

What Pro-C does have is copy-protection. It allows only three installations. Ugh. If copy-protected programs in general are to be avoided, copy-protected programming tools are to be shunned altogether. I despise them.

Raney's Screen Editor

Version 2 of Dennis L. Ramey's Screen Editor is potentially the beginning of progress towards a nice little screen-oriented screen generator. Right now it's a moderately usable, unpretentious program written in Turbo Pascal. It seems to generate decent code for Turbo Pascal or BASIC, but is only one step on the path to doing well for C.

The C code in Listing 1 was generated by Screen Editor, and all but one of the functions call Bob Pritchett's C Spot Run function library for Version 3 or 4 of Microsoft C. You'll need to either get C Spot Run or translate the calls so they match whatever other library you own. `scanf()`, in Microsoft's standard library, is the one way to get user input right now. It's unsophisticated but workable.

One major flaw in Screen Editor (ignoring the awful spelling of its screen messages) is that it does not know how to coexist with a high-resolution monochrome system. You can produce code for programs to run on that kind of system, but don't try running Screen Editor itself on it: more often than not you'll lock the system. A sure way to do that is by pressing `CTRL-H` for help. Don't.

Both Raney's Screen Editor and Pritchett's C Spot Run are distributed as shareware. You can download copies of these packages free from local bulletin boards and experiment for a while. Should you decide to use them, registration costs \$25 for Screen Editor and \$50 for C Spot Run. If you want to get some idea of whether you need a screen generator, these two shareware programs seem a reasonable place to start.

Dan Bricklin's Demo Program

Yes, the name of this program really is Dan Bricklin's Demo Program, not just Demo. If I could have written this package I too would make darned sure to put my full name on it proudly, and I'd be annoyed at anyone who committed abbreviation upon me. A particular reason for my annoyance would be that I wouldn't want the product of my exclusive genius confused with an imitation written in Turbo Pascal and sold by Shamrock Microsystems of Indianapolis for \$35. The imitation is sometimes distributed as apparent shareware under the Demo name, but turns out to be just a crippled demo itself.

Imitations sometimes turn out to be better than the original real thing, but not this one. If I were Dan Bricklin, I wouldn't want my original real thing—written in Microsoft C and assembler for speed, and sold as the only product of my one-man company for just about twice that of the imitation—confused with the knockoff.

I'm not being in the least hyperbolic when I attribute genius to both Dan Bricklin and his demo program. You'll remember Bricklin as the man who invented Visicalc, the first spreadsheet software. It first ran on those cute little Apples, which seemed to have no reason for being except to run games. Visicalc showed many people that microcomputers could be a serious business tool.

Now, after what looks to have been some well-earned time to catch his breath, eat the lotus, or whatever cliche might be more appropriate, Bricklin has tossed out a Visicalc for programmers. Dan Bricklin's Demo Program and Visicalc are products of the same vision.

Dan Bricklin's Demo Program allows a programmer to play "what if" with a program's user interface. It's a sketch pad on which you can work out the program's appearance screen by screen. Since everything is illusion at this stage anyway—only screen images with none of the painstaking code normally required to create them—changing anything at all is easier than pie. You move your cursor to where you might want a design element, then type your text or mark the area as the beginning of a block. The end of the block is marked wherever you next place your cursor.

You can move blocks, center them, copy them, delete them, save them, frame them, fill them with color or other display attributes, layer them, or do just about anything else you can think of to do with a rectangular area. I don't think you can do circles or ellipses—at least I can't.

Dan Bricklin's Demo Program includes a runtime module that will display a series of screens according to a script you write. This enables you to animate the screens as if they were in a running program. You can even simulate keyboard strokes that would activate menu choices. Why, you could even orchestrate the entire sequence to produce a self-running demo!

Dan Bricklin's Demo Program therefore seems to me an essential tool, not only for a professional programmer but for any programmer concerned with the relationship between how a program looks and how it works.

I must confess that Dan Bricklin's Demo Program is making me believe that a right understanding of that relationship is pretty close to what I have been trying to teach students about writing: you can't separate content from its expression. Dan Bricklin's Demo Program has helped me

approach that understanding by making it simple for me to experiment visually. I realize now that you develop sensitivity to program design in just the same way you learn writing: by revising and editing. "How do I get to Carnegie Hall?" "Practice!" Dan Bricklin's Demo Program makes revision, editing, and practice so easy that they're really fun.

One game that makes Dan Bricklin's Demo Program much more fun than arcade games or other timewasters is doing dummies of your favorite software. There's a RAM-resident screen-capture program included so you can take snapshots of a program like Lotus 1-2-3 in action. Then you run Dan Bricklin's Demo Program, import the captured screens, and edit them into the desired sequence. I've done parodies of a few bestsellers that way. More important from a practical point of view, I've been able to study their architecture so I could learn from them. It's not that I intend writing a commercial program: I merely want to understand better what I do and, bit by bit, to do it better than the time before. Call it "pride."

The weakest part of Dan Bricklin's Demo Program seems to me to be what happens once you get your program working just right visually. Among the several output choices selectable from a nice set of pop-up menus is a print-to-disk option that allows translation of the screens into a text file you can massage with a text editor. That way you can write a file with no character mapping, in which case you get straight ASCII text, or you can write it with control characters mapped for inclusion in Pascal or C source code. The trouble is that the mapping for C results in text characters accompanied by octal code that's supposed to represent their attributes. It's pretty hard to reconstitute from them those beautiful screens made so easily within Dan Bricklin's Demo Program itself.

Nevertheless, if there were an Oscar for Best Program of the Year, Dan Bricklin's Demo Program would get my vote. Given a choice between the genuine article at \$75 and a half-priced imitation, you'd be silly to pass up the real thing.

C-scape

It's satisfying when things fall into place. The more I became entranced by the ease with which I could use Dan Bricklin's Demo Program to work out screens for my programs, the more reluctant I became to code those screens. Dan Bricklin spoiled me the way Cinderella's fairy godmother spoiled her one night. I now can empathize with how she felt that moment her coachmen became mice once again. "Rats!" thought I, wondering why someone couldn't invent a program to translate those intricate Bricklin screens directly into C source code. Of course, someone did.

The program is C-scape. You need Dan Bricklin's Demo Program and either the Lattice or Microsoft C compiler. (There's a different version of C-scape for each, so be specific when you order.)

Once you have your screens designed just right with Dan Bricklin's Demo Program, you print them to disk files as text according to C-scape's requirements. The requirements are not quite what you might expect, so pay attention to Appendix D in the C-scape manual: the characters are not mapped, and display attributes are segregated from the characters themselves in a separate section at the end of the text file. These are choices you make from a menu when you print to disk, so you needn't understand anything about why you're making those choices. The C-scape instructions are clear enough, so I can't imagine anyone's having trouble with them. And the way to effect them in Dan Bricklin's Demo Program is easy enough, so I can't imagine anyone's having trouble there either.

In fact, the process is almost too easy. There's a small translator program in C-scape called Demo2c: run it with the name of your Bricklin text file as the only argument (DEMO2C MATTHEW, for example) and in a few seconds—seconds!—you get the C source code (MATTHEW.C).

Things can get pretty complicated here, mostly because the C-scape manual doesn't really give a good overview of

SOFTWARE FOR ALL ZENITH COMPUTERS

**H8, H89, H100,
H150, H248**

PERFECT PRINTER

Electric typewriter (2 modes) PLUS 7 print functions.....\$19.95

PAYCHECK

Payroll program with multiple wages, salaries, tips, federal taxes, and MORE! Prints checks, quarterly/annual reports.....\$39.95

MT ACCOUNTANT

Organizes financial data by account or date. Easy to use!.....\$19.95

PERFECT FUNDS

Manage your money market and stock funds! Tracks performance of each fund and your entire portfolio. A must for today's market!.....\$29.95

PERFECT MONEY

Calculates loan costs, payments, interest, and balloon payments.....\$19.95

ORDER NOW by writing or calling:

Micronics Technology

(904) 897-4257

449 Barbados Way

Niceville, FL 32578

We accept checks, Visa, MasterCard.

Shipping and handling: \$3.

Florida residents add 5%

Circle #208 on Reader Service Card

either the whole process or each stage in it. It's programmer's documentation more than a user's manual. Like the rest of that breed, everything comes clear in a flash once you've banged your head against the wall sufficiently.

The idea is that C-scape is the front end to your program. That front end employs its own functions (expanded in supplied headers and libraries) for comprehending and manipulating the keyboard and screen up until the time the user has pushed all required data through the openings you've provided in your input screen. Then your program proper takes the data and does what you want with it.

Say you write a mailing list program. You would use Dan Bricklin's Demo Program to build its menu, data-entry, and help screens—the face seen by your user. When the look and feel are just right, you use Dan Bricklin's Demo Program to make the text files C-scape requires. Then you put Dan Bricklin's Demo Program away and start using C-scape.

You run Demo2c across each of the text files to transform them into C source code so that each is a program module. Then you put away the text files too.

Your program modules now are set to return the values that determine which choice the user has made at each menu and what data the user has input in each field of each input screen. The action your program takes in response to each menu choice and what it does with the data input are up to you: you write the code that determines all that. In other words, you're solely responsible for the data processing that goes on behind the scenes you set with Dan Bricklin's Demo Program and C-scape.

C-scape right now is not yet perfect, but it's much, much better than merely pretty good. I confess to feeling a little ungrateful being even mildly critical right now: Dan Bricklin's Demo Program itself has been out only nine months as I write this, and C-scape is several months younger. It's still being polished so actively that a few days ago I got a free upgrade from 1.0 to 1.0A.

Now that I have said enough statesman-like things to maintain my well deserved reputation for fairness, I can say that there are three areas that need work. One is that the finished programs are bloated a bit too much by C-scape. The functions can be slimmed. The Oakland Group, makers of C-scape, knew so without my telling them. Version 1.0A is evidence that they're working in this area.

Another area for improvement is the manual. Big and detailed it is, but a manual it isn't. I think it needs wholesale recasting and rewriting before it won't give you a headache—although, as I've said, once you break the barrier a great deal falls into place almost all at once. Fortunately, there are some really nice

examples included in the C-scape package: study them in conjunction with the manual and the breakthrough will take place sooner than it might otherwise.

The third area in which I would like to see something done might just be a matter of my own preference. I'd like the ability to make display-only screens that don't have to carry the apparatus for data entry: not all programs require it, and even those that do require it don't for every screen.

Those are really minor cavils about a package that does really nice work. My general opinion is that Dan Bricklin's Demo Program without C-scape is like Visicale would have been without a way to print.

Zview

Data Management Consultants' Zview is the most sophisticated and complete screen generator I've seen so far. It's a thoroughly professional set of tools in one matched package.

Zview's screen editor is Zpaint: you use it to lay down the screen and any data

The nine functions peculiar to Zview speak to sleekness rather than sparseness.

fields. The screen can have all the pizzazz you might like—windows, boxes, blink, and almost anything else you can imagine—along with quieter, useful touches such as an easy help screen to go along with each major screen. You can designate display-only fields, input fields, and header fields, and you can set a security level (the significance of which you decide) for each screen. Zpaint produces a screen image file that can be recalled and modified as often as needed.

The code generator is Zfield: you use it to designate the data types acceptable in each field (all C data types are supported), check ranges for data entry (for example, you can prohibit lowercase or uppercase alphabeticals), and set relationships among fields. When you've finished with Zfield, it generates the C code.

The code generated by Zfield calls, in addition to the standard functions, only nine functions peculiar to Zview. Those nine functions speak to sleekness rather than sparseness: the small number of functions results in finished programs much smaller than are likely to be produced with a less polished screen generator. Those functions are expanded at link time with Zview libraries coordinated with the Microsoft, Lattice, or Aztec compilers. (You need to specify your

compiler when ordering.)

There are only a few things I don't like about Zview.

One thing I don't like is that Zview expects to find its support modules in a subdirectory named ZSCREENS. The subdirectory name and the path to that subdirectory are hardcoded into ZPAINT.EXE and ZFIELD.EXE. It seems that more and more packages want to be installed in their own subdirectories nowadays. No, I won't be told what to do. I don't like it.

I eventually used a disk editor to remove the hardcoded pathname, and now Zview works the way I want. I keep its programs on a single floppy diskette and I copy them onto my hard disk whenever and wherever I need them. I wish Data Management Consultants would omit the hardcoded pathname from future versions of Zview and let me live comfortably in my own untidy mess.

The second thing I don't like is that Zpaint and Zfield accept only uppercase arguments on the command line. To create a screen file, therefore, only "FOO"—not "foo" or "Foo" or anything else with lowercase letters—will do. It's maddening and unnecessary because the case of arguments is easily forced. More maddening still is that when you use lowercase the Zview modules die with a terse INVALID PARAMETERS—surely the most unhelpful error message conceivable. Ultimately maddening is that the manual nowhere points out the essential requirement for uppercase. In fact, I had thrown Zview in my discard pile after a few hours of trying could not get me past the INVALID PARAMETERS message. What eventually cracked the case for me was one last try after I got to wondering why the error message was completely uppercase. "Could it be . . . ?" It was.

The third thing I don't like is the manual. I like Zview itself so well that I'll be charitable and say little more than that the existing manual ought to be shredded and an entirely new manual issued. I'm not talking about misspellings and such. I'm talking about chaos. Here's a one-paragraph example for you to chew on:

The fact that ZPAINT.EXE and ZFIELD.EXE except a full DOS path when loaded, you can load either of them from any directory you happen to be in. This is the reason for the "zscreens" directory.

What I had to do, after I caught on to how to kickstart Zview and then found it well worth the trouble to figure out, was to make a set of notes that I now use instead of the manual. When I find the time, I'm also going to make a reference card. The Zview manual makes that effort necessary; the rest of Zview makes that effort worthwhile.

My three strong objections notwithstanding, Zview stands out from the crowd. Not the least of its notable features

is Data Management Consultants' commitment to supply registered owners with free upgrades. I got one.

Ordering Information

Microsoft C Compiler (Version 4), \$450.

Microsoft Corp.
16011 NE 36th Way
Redmond, WA 98073-9717
800/426-9400; in Washington and Alaska, 206/882-8088; in Canada, 416/673-7638
Call or write for dealer nearest you.

db-Vista (Version 2.10), single-user, \$195; multi-user, \$495.
Raima Corporation
12201 S.E. Tenth Street
Bellevue, WA 98005
206/747-5570

ISAM Driver, \$40. BTree Library, \$75.
Softfocus
1343 Standbury Drive
Oakville, Ontario, Canada L6L 2J5
416/825-0903

Btrieve, \$245.
SoftCraft, Inc.
P.O. Box 9802 #917
Austin, TX 78766
512/346-8380

Pro-C (Version 1.0), \$195.
MAJIC Software
224 South Salisbury, Suite C4
West Lafayette, IN 47906
317/743-0610

Screen Editor (Version 2.0), \$25.
Dennis L. Raney
PC Applications
2612 Castle Drive
Blue Springs, MO 64015

C Spot Run, \$50.
New Dimension Software
23 Pawtucket Drive
Cherry Hill, NJ 08003
609/424-2595

Dan Bricklin's Demo Program, \$74.95.
Software Garden
P.O. Box 373
Newton Highlands, MA 02161
617/332-2240

C-scape (Version 1.0A), \$149.
Oakland Group
675 Massachusetts Avenue
Cambridge, MA 02139
800/233-3733, 617/491-7311

Zview (Version 1.6), \$245.
Data Management Consultants
5325 South Valley View Blvd., Suite 7
Las Vegas, NV 89118
800/423-0930, 702/798-5910

Have you considered writing about your Heath/Zenith computer?

Whether you're a hobbyist, programmer, writer, or hardware specialist, you may be carrying around an idea for an article. Think about it....

To get your knowledge to our readers, you mainly need:
A Heath/Zenith computer • A willingness to write about it

We're looking for material—human-interest stories, in-depth technical features, how-to articles, hardware construction descriptions, product reviews, tutorials, program descriptions—dealing specifically with Heath/Zenith equipment and users. We cover *all* Heath/Zenith machines—from the '8 to the most recently announced machines. If you're using a '150, for instance, we'd like to see an article on anything of particular interest to the user of a *Zenith* IBM-compatible computer.

Articles for *Sextant* should be written in such a way as to be interesting to the advanced reader, but still accessible to the beginner.

In the way of programs, we're looking for utilities, games, and other applications that could be used by anyone who owns a Heath/Zenith computer. Programs themselves should be kept short—a BASIC program of 150 lines, for instance, is getting long.

Long programs are of interest if the focus of the article is on the design and concepts behind the program. These should be "educational" articles—teaching us how a particular process works.

Hardware construction articles are also of interest—if you've made an enhancement to your system that you'd like to write about, we'd like to read it.

And certainly, if you've purchased a newly released product and have tested it on your machine, we'd like to share your perspective on it with our readers.

We could go on—but rather than that, **send us your ideas**. We know a lot of you out there are doing some pretty interesting things that the rest of us would like to hear about.

We pay from \$50 to \$100 per *Sextant* page of printed copy.

Send your idea, abstract, outline, rough draft, or article (no disks, please!) to:

Charles Floto, Editor
Sextant
716 E Street, S.E.
Washington, DC 20003
(No phone calls, please!)

Writing Hardware-Independent Terminal Emulators for CP/M and MS-DOS

Here's how to turn your computer into a dumb terminal, whether you're using an H8, H/Z89, or H/Z100. You won't need to get bogged down with the hardware, and you may get to know your operating system a little better.

William S. Hall

I would like to give you a terminal-emulator program.

Now, terminal-emulator programs with elaborate features are practically free today; what I will offer here will hardly match the quality of such software. Rather, what I really want to show you is how to write a program to turn your computer into a dumb terminal without your having to know very much about the details of your hardware environment.

You can use the information to improve your understanding of your computer. At the same time, you will have a program that you can expand and tailor to your needs if you have such ambitions.

In the first part of the article, I will discuss CP/M's IOBYTE. The information from that discussion will be used in producing a program capable of running on the H8, H/Z89, or H/Z100 under CP/M. It will run without any changes, even though the hardware configurations of the H8/89 and Z100 are vastly different.

In the second part, I will provide similar programs for the Microsoft Disk Operating System (MS-DOS)—both MS-DOS 2 and the Zenith Disk Operating System (Z-DOS), Zenith's implementation of MS-DOS version 1. At the same time, I will show you how to use the resources of the operating system to accomplish the tasks required.

I should emphasize the importance of utilizing the operating system to achieve our goals. In multi-user and/or multi-

tasking operating systems (such as Concurrent CP/M, Xenix, Windows, and MP/M), it is bad practice for a program to bypass the operating system in order to deal with the computer and peripherals directly. On most mainframe computers, it's not even possible. Even in single-user systems, such procedures should be used only when absolutely necessary. They de-

greatly reduce the cost of software development. In addition, programs written in high-level languages are much easier to understand and modify.

Terminal emulation

What must a terminal emulator do? In "pseudo language," a possible form for such a program is:

```
repeat forever
  if there is a character at the terminal
    write it to the communications port;
  if there is a character at the
    communications port
    write it to the terminal.
```

Since all programming languages offer some way of reading and writing to the console, and since most provide a method for reading and writing to a communications port where a modem can be connected, where is the problem?

The difficulty is the assumption behind each of those "if" clauses: a terminal emulator must be able to check the *status* of the terminal and the communications port without waiting for a character to appear. By "status" of the port, I mean: is there a character waiting to be read?

If no character is present, we want the program to move on immediately. Clearly, we do not want it to wait for something to be typed at the terminal while ignoring a stream of data entering the communications port. Similarly, we must be able to look at the status of the communications port and, if there is nothing there, to go back to check the terminal.

Most calls to the console (whether *readln* in Pascal, *getchar* in C, etc.) do not provide the status information we need; instead, they simply wait for a character to be typed.

*I want to use
system resources
entirely to do
the job.*

feat the uniform interface provided by the operating system's services.

Part I: The CP/M Version

I will present two CP/M versions of the program, which I call simply TTY. One version is in C/80, The Software Toolworks' excellent implementation of the C language. Of course, not everyone has C/80, but all versions of CP/M come with an assembler and an editor. So, I will also give the same program in an .ASM version.

It is interesting that it has become possible to write in a high-level language programs that are traditionally done in assembly language. Today the trend is to use a language such as C or Pascal for most of a program, and assembly language only to provide any needed hardware interfaces. With the rapid changes in central processing units (CPUs), such an approach can increase portability and

William S. Hall, formerly editor of Mathematical Reviews at the University of Michigan, is now with AT&T Information Systems in Freehold, New Jersey.

In addition, most of these console calls intercept and act upon control characters commonly used in editing, printer control, and flow control. If we are talking to a remote computer, not being able to send and receive certain characters may be quite inconvenient.

The implication is that we must use some resource of the operating system, or we must learn the hardware well enough to make the status calls directly to the ports. Here, I want to emphasize the former.

CP/M provides several calls for operating-system services for the console. These system calls include the usual ones that let the CPU read the keyboard (system call 1, console read) and write to the screen (2, console write). Another provides the status of the keyboard (11, get console status).

Regarding input/output (I/O) between the CPU and the console, call 10 provides buffered input to the CPU from the keyboard; direct keyboard input to the CPU and output from the CPU to the screen is allowed by 6. System call 6 passes any character to or from the program without any special action by the operating system; not only that, it also checks status. It is the one to use in our simple application.

The problem remains how to accomplish the same thing for the communications device.

A glance at the list of function calls and their definitions in the CP/M interface guide shows that one can read from logical devices called CON: and RDR:, and write to PUN:, LST:, and CON:.

Each of these logical devices can be connected to various physical devices. (See Table 1.) For example, the CON: device can be attached to the video terminal (CRT:), a teletypewriter (TTY:), a user-defined console (UC1:), or to the batch (BAT:) device. (The latter will be very important to us; the explanation below should clarify its use.)

The connection between logical and physical devices is governed by an 8-bit value called the IOBYTE. We can associate a logical device with one of four possible physical units by appropriately defining each of IOBYTE's four pairs of two bits.

IOBYTE is found at location 0003 in memory; its value can be examined by using a program such as DDT that lets you examine the actual binary code.

Table 1 gives the options possible for different values of bits 0-7 of the IOBYTE. For example, if bits 6 and 7 are 01, then whenever you write something to the printer (LST:), it will be routed to the CRT: physical device; on the '89, this is the screen. If the values are 10, then the output goes to LPT:; this is the printer port, number 340 octal (340Q) on the '89.

In Heath/Zenith CP/M, the center four bits (bits 2-5) are usually 1010. This simply means that a function call 3 (input from the reader) will try to find a

Logical Devices	LST:	PUN:	RDR:	CON:
Bit-pair values,	00	TTY:	TTY:	TTY:
followed by the	01	CRT:	PTP:	PTR:
physical devices	10	LPT:	UP1:	UR1:
associated with them	11	UL1:	UP2:	UR2:
				UC1:

Table 1. In CP/M, the bit-pair values of the IOBYTE control the mapping of logical devices to physical devices. The table above shows which physical device is specified by the associated bit-pair value in the IOBYTE. For example, an IOBYTE value of 01100100 binary (C4 hex) would map LST: to CRT:, PUN: to UP1:, RDR: to PTR:, and CON: to TTY:.

character at RDR:’s physical device (URI:, which is at port 330Q on an '89). Function 4 (output to the punch) will write a character to PUN: (UPI:, also at port 330Q).

Unfortunately, no status checks are possible with function calls servicing RDR:. In fact, the only logical device that provides status is CON:, and this can be mapped only to physical devices TTY:, CRT:, BAT:, or UC1:.

The device of interest here is BAT: (batch). With BAT:, console I/O takes place through the current physical devices of RDR: and LST: instead of through CRT:. RDR: takes the place of keyboard input, and output goes to the current LST: physical device instead of to the screen.

So, if the two least-significant bits of the IOBYTE are set to 10, then CON: is mapped to RDR: and PUN:. Every system call which services CON: (namely 1, 6, and 10) will end up reading whatever physical device is mapped to RDR:. Any console write call (functions 2 and 6) will go to whatever is connected to the LST: device. As we have seen above, LST: can also be routed physically to the screen, CRT:.

The CP/M manual states that if we call function 6, it will return with a null if CON: has no character for us; it will return with the character itself if one is ready. So if we use function 6, the status problem is completely solved.

Now we can see how to implement the TTY program defined in pseudo-code above: by using IOBYTE switching. Before we check the console for a character, we use function call 8 (set the IOBYTE) to ensure that the usual configuration holds. The binary value 10101001 is used during normal CP/M operations: LST: is connected to the printer LPT:; RDR: and PUN: are attached to URI: and UPI: as mentioned above; and CON: is connected to the CRT:. After setting the IOBYTE, we can use system call 6 to check the console status.

When we wish to read the communications port, we will change the IOBYTE to 01101010. This will connect CON: to BAT: and LST: to CRT:. Using an identical function 6 call will then do for the communications port exactly what it does for the console during normal operations. This dynamic switching of the IOBYTE is very

neat, and it does not seem to be widely known.

Thus, we can write a program that is nearly portable across most CP/M machines without knowing anything other than the proper values to use for normal and batch modes. For the H8 or '89 and the Z100 running CP/M, the programs are identical. (An obstacle, however, to portability to other CP/M machines is that not all support the IOBYTE; if they do, the meaning of the bit pairs may depend on the hardware.)

The C program

The greatest similarity to the pseudo-code given above is provided by TTY.C. Let’s take a look at the source code, shown in Listing 1.

The customary `printf` function for formatted output is not a standard function in C/80. So, we first have to include the `printf.h` file, which C/80 uses to support `printf`. This function is used only to display the banner when the program starts to execute.

The various `defines` are not necessary, but they let me keep the rest of the program simple and make it easy to modify and understand.

The main interest centers on the `while (TRUE)` loop.

Notice that the first statement of the loop sets the IOBYTE to normal via a system call to CP/M. The console is then read with a direct console call (6). If a null is returned, the program falls through to check the modem port.

If there is a character present at the console port, it’s checked to see if it’s the control-backslash (CTRL-\)—the exit/abort character—saying we should exit. If the character is not the CTRL-\, it is sent to the communications port by means of another system call 4 to the PUN: device. Because of the way the IOBYTE is set, the output is sent to the communications port.

The program then sets the IOBYTE to batch mode and makes the identical function 6 call to the CON: device. Batch operation ensures that the call is actually directed to the communications port. If there is an incoming character waiting at the port, the program makes a normal write call to the console device with

putchar. Then the main loop repeats.

In batch, the output must go to the logical device LST:. But we have set the highest two bits in the IOBYTE to connect LST: to CRT:. Thus, the output appears on the terminal.

The ASM program

The assembly language version of TTY is given in Listing 2. This program's operation is basically the same as the C program above. I have tried to hide some of the details via subroutine calls. But, as you can see, we never have to know very much about the actual port values or how to make a real "hardware" read or write to the communications port.

Using the programs

If you decide to test the C/80 version, follow C/80's normal procedures to generate a program from the source code I have provided.

If you use the assembly language version, assemble it with the command ASM TTY. Fix any errors that may have crept into the program while it was being typed in. When it assembles with no errors, load it with the command LOAD TTY. You can then execute it with the command TTY.

I assume that you have already configured your machine so that the communications port has the correct baud setting, word size, parity, etc. You are now ready to try communicating with the outside world.

Some remarks

As TTY stands, its sequence of operations is not quite the best, since you cannot use your printer with it.

That is because the change to BAT: ties up the LST: device. I found the program I've presented more symmetric, but an equally good alternative is to change the IOBYTE only when reading the communications port, and then to switch back to normal before issuing putchar. In this case, only the lowest two bits of the IOBYTE need be changed. For Heath/Zenith CP/M, the new value will be 0AA hexadecimal. The printer can remain attached to LPT:.

Another improvement would be to begin the program by saving the value of the currently used IOBYTE. Also, the bit-manipulation operators of C could have been used to modify the IOBYTE value during program execution. I have not done either here, since it would simply clutter the program. (But it is good programming practice to preserve the user's environment upon entry and restore it upon exit. Not all systems may regard my choice of IOBYTE value as "normal.")

It has been difficult to locate the exact numbers for the IOBYTE in Heath/Zenith documentation. A very useful source, which I eventually had to use, is the *CP/M Programmer's Reference Guide* by

```
/*
Simple terminal emulator illustrating the use of the I/O byte

Written by: William S. Hall
N24A Pine Tree Boulevard
Old Bridge, NJ 08857

Language: C/80 from the Software Toolworks

Date written: 01/21/84

Usage: Use a dial-up modem connected to another computer
or in analogue loop-back mode. Run TTY, communicate,
exit with a control-\.

*/
#include <a:printf.h> /* needed to print a banner */

#define TRUE 1 /* Exit character */
#define FS 28 /* Modem output */
#define PUNOUT 4 /* CP/M direct I/O call */
#define DCONIO 6 /* CP/M set IOBYTE call */
#define SETBYTE 8 /* Want to read char */
#define RDCHAR 0xFF /* Normal I/O byte (LST:=LPT:, PUN:=UP1,
RDR:=UR1, CON:=CRT: */
#define NORMAL 0xA9 /* Batch I/O byte (LST:=CRT:, PUN:=UP1,
RDR:=UR1, CON:=BAT: */

main()
{
    int c;

    printf("Welcome to TTY. Type control-backslash to exit\n\n");

    while (TRUE) { /* Do forever */
        bdos(SETBYTE, NORMAL); /* Set for normal */
        if ((c = bdos(DCONIO, RDCHAR)) != 0) /* See if char at console */
            if (c == FS) /* Check for CTRL-backslash */
                exit(); /* Exit program */
            else
                bdos(PUNOUT, c); /* Send to modem */
        bdos(SETBYTE, BATCH); /* Set for batch */
        if ((c = bdos(DCONIO, RDCHAR)) != 0) /* See if char at modem */
            putchar(c); /* Send to console */
    }
}
```

Listing 1. TTY will turn your computer into a dumb terminal. The simplest version of the program is written in the C/80 language to run under CP/M. Since it makes use of the operating system rather than dealing directly with the hardware, the same version of TTY will run on the H8, H/Z89, and H/Z100.

Listing 2. The assembly language version of TTY for CP/M is structured similarly to the C/80 version (Listing 1). The use of subroutines makes the lengthy assembly language version easier to follow. Note that this version of TTY, like the C/80 version, lets you avoid the need to know much about hardware-specific details.

```
; Terminal Emulator using the IOBYTE

; Written by William S. Hall
; N24A Pine Tree Boulevard
; Old Bridge, NJ 08857

; Usage: Run TTY. Exit with Control-\.

CR EQU 13 ; carriage return
LF EQU 10 ; line feed
BDOS EQU 0005 ; jump to BDOS
PUNOUT EQU 4 ; output to PUN:
DCONIO EQU 6 ; CON: I/O
SETBYT EQU 8 ; set the IOBYTE
PRNSTR EQU 9 ; print a string to the terminal
FS EQU 28 ; return-to-system character, control-\
NORMAL EQU 10101001B ; normal value of IOBYTE
BATCH EQU 01101010B ; batch value of IOBYTE with LST:=CRT:

ORG 100H ; start here

START: LXI D,BANNER ; print a banner
        MVI C,PRNSTR
        CALL BDOS

; main loop
```

```

TTY:    MVI    E,NORMAL      ; get normal value of IOBYTE
       CALL   SIOBYT      ; set the value
       CALL   RCONS       ; see if a character is there
       CPI    FS          ; control-\ exit
       RZ
       ORA    A           ; null, no char
       JZ    RDR          ; no char, so check modem
       CALL   WPORT       ; character, so write it out
RDR:    MVI    E,BATCH      ; get the batch value of IOBYTE
       CALL   SIOBYT      ; set it
       CALL   RCONS       ; see if char at modem port
       ORA    A           ; null, no char
       JZ    TTY          ; no, so back to tty
       CALL   WCONS       ; write the character to the cons
       JMP    TTY          ; do it again

;      end of main loop

;      SIOBYT - set the I/O byte

SIOBYT: MVI    C,SETBYT    ; set the value
       CALL   BDOS
       RET

;      RCONS - read the CON: device.  Return char or 0 in A.

RCONS:  MVI    C,DCONIO    ; set the value
       MVI    E,0FFH
       CALL   BDOS
       RET

;      WCONS - Write to the CON: device.  Character in A.

WCONS:  MOV    E,A
       MVI    C,DCONIO    ; set the value
       CALL   BDOS
       RET

;      WPORT - write to modem port.  Character in A.

WPORT: MVI    E,A
       MVI    C,PUNOUT    ; set the value
       CALL   BDOS
       RET

BANNER DB    'Welcome to TTY.  Type Control-\ to exit.',CR,LF,CR,LF,'$'

END    START

```

Sol Libes, the editor of *Micro/Systems*. The BIOS listing for CP/M 2.2.04 also proved necessary.

You can verify most of the figures in Table 1, though. Change the IOBYTE by using STAT or CONFIGUR to alter the logical-to-physical device mapping, then look at the results by running DDT and displaying page zero of memory with the command D0. The IOBYTE is at location 0003. (But, of course, if you switch to BAT, you have already lost your console, and the exercise is academic!)

With all these ideas as a basis, you can now think about writing a more versatile program—one that will allow both simple capture and protocol-governed file transfer as does Kermit or the XMODEM protocol.

The C program provides a basis for considerable expansion relatively easily; complicated aspects such as file openings and closings, buffering, and command parsing are easily handled in C. It is more troublesome in assembler, but, of course, that language is more flexible and powerful if you must get to every aspect of the hardware.

Both programs illustrate how system resources can be used to fulfill program-

ming needs. They should be easy to understand and to carry to other machines. And looking at the programs, you can see that C/80 can provide many of the calls normally implemented in assembler, but at the cost of a much larger object program.

In fact, there is only one real shortcoming to using C/80 in this application: the lack of a convenient function to read and write to an arbitrary port number. However, this would be necessary only if you wanted to be able to change baud setting, parity, and word size dynamically. At that point, though, you would have to become familiar with your computer's hardware, unless such system calls have been provided for you. Such calls exist in MS-DOS.

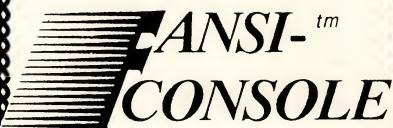
And some conclusions

Before I close this half of the article, let me note that once you have TTY running you have a kind of rudimentary file transfer, at least for sending from the micro to a host computer.

After using TTY to log on to the remote machine, call up its equivalent of the copy command to copy from the terminal to a disk file. (On a DEC-20, for example,

**MAKE YOUR PC
SEEM LIKE AN AT!**

**MAKE YOUR AT
SEEM LIKE A
DREAM MACHINE!**



*The Integrated Console Utility™
FAST, POWERFUL
ANSI.SYS REPLACEMENT*

For the IBM-PC, AT, and clones

New Version 2.00 is MUCH FASTER!
Now blink free scrolling on CGA!

Now uses EMS/EEMS for Scroll Recall
New Menu Program for Changing Options

**GET A BOX FULL OF UTILITIES!
MAKE LIFE EASIER FOR ONLY \$75!**

- Speed up your screenwriting 2-6x
- Extend your ANSI.SYS to full VT100
- Add many more escape sequences
- Scroll lines back onto screen
- Save scrolled lines into a file
- Add zip to your cursor keys
- Free your eyes from scroll blinking
- Easy installation
- Get a 43 line screen w/EGA
- Get a 50 line screen w/CGA
- No more annoying typeahead beep
- Prevent screen phosphor burnin
- Control many programs' use of color
- Generate breakpoints from keyboard
- Shorten that annoying bell
- Over 50 other useful options

*"The psychological difference is
astonishing"*

—Lotus June 85 pg 8.

*"So many handy functions rolled into
one unobtrusive package"*

—PC-World Feb '86 pg 282.

*"The support provided by the
publishers is extraordinary"*

—Capital PC Monitor May 86 pg 25.

*"...the best choice for improving your
console..."*

—Capital PC Monitor June 86 pg 26.

460p Manual (w/slip case) & disks \$75.

**Satisfaction Guaranteed!
Order Yours Today!**

HERSEY MICRO CONSULTING

Box 8276, Ann Arbor, MI 48107
(313) 994-3259 VISA/MC/Amex

DEALER INQUIRIES INVITED



Circle #168 on Reader Service Card

the command is `copy tty: filename`. On a DEC-10, the command is `copy filename=tty:..` Exit from TTY with a `CTRL-\` and give the CP/M command `PIP UPI:=FILENAME`.

On that command, the file will be read to the user punch 1, which is the physical device connected to the communications port, and hence sent to the host. When the transfer is completed, run TTY again and close the file on the remote host. On a DEC-10 or DEC-20, this means simply typing a `CTRL-Z`.

You will find the file, but it will probably contain an extra line for each line in your text. This happens when the remote host expects you to send just a carriage return to indicate the end of a line; your text file, however, will also be sending a line feed after the carriage return. On some hosts this may not matter, and you may be able to arrange things to make this mode of operation satisfactory. Alternatively, you can write a filter in C to remove the line feed from your file.

It is possible to read a file from the remote machine with `PIP`, using the command `PIP FILENAME=URI:(user reader 1)`. Unless you can put the remote computer to sleep for a while, however, you will probably not have enough time. Some of the file would have already been sent in the time needed to ask the remote to type the file, exit from TTY, and set up `PIP`.

Part II: The MS-DOS Version

Let's continue our discussion of TTY.

Again the aim is to achieve our goal without depending on a detailed knowledge of the machine's hardware. I want to use system resources entirely to do the job. I have elected to use 8088 assembly language for this version since not all of us have a high-level language on our computer, but everyone with Z-DOS has an editor, assembler, and linker included with it. (Those of you with MS-DOS version 2 will need the Programmer's Utility Pack if you do not already have an assembler.)

I had hoped to show you how to do this not only on the Z100 but also on the Z150. Unfortunately, the serial-port hardware environment on the latter machine is not nearly as good as on the Z100. Whereas the Z100 version of MS-DOS provides a communications-port buffer, the Z150 version does not.

Thus, the performance of my simple program is so poor that it is not really worth the effort to implement it. To avoid dropping characters, I would have to write a suitable buffer and a routine to handle the interrupt signals sent by the port to the CPU. So I must confine this discussion to the '100.

I would also like to show you how to use the special functions that the Z100 version of MS-DOS provides for the character devices—those devices that can deal with a character at a time, namely the PRN: (printer), CON: (console), and AUX: (communication) devices. They are

```

        title  ftty - fast terminal emulator for Z100
;
; Usage: Run FTYY.  Exit with control-backslash.
;
; Limitations: CTRL-\ cannot be sent to host computer.
;               No file transfer is possible.
;
; Written 03/30/84 by William S. Hall
;           N24A Pine Tree Boulevard
;           Old Bridge, NJ 08857
;
; .XLIST                      ; do not list these files
INCLUDE DEFASCII.ASM          ; Zenith-supplied file
INCLUDE SEGMACRO.MAC          ; segmentation macros
INCLUDE DOSMACRO.MAC          ; MS-DOS string display
; .LIST                         ; restore listing
;
; SSEG   100                   ; make 100h stack segment
;
; DSEG   PARA                  ; start data segment
BANNER DB  'Welcome to FTYY terminal emulator.  Type CTRL-\ to exit.'
DB  CC_CR, CC_LF, '$'          ; end data segment
;
; PSEG   PARA                  ; start code segment
MAIN  PROC  FAR                ; main program
      FAR
      EXTRN CD_AUXIN:NEAR          ; external programs
      EXTRN CD_AUXSTAT:NEAR
      EXTRN CD_AUXOUT:NEAR
      EXTRN CD_CONIN:NEAR
      EXTRN CD_CONSTAT:NEAR
      EXTRN CD_CONOUT:NEAR
;
; START:
      PUSH  DS                  ; save the data segment
      MOV   AX,0                 ; now add the offset for return
      PUSH  AX                  ; stack ready to return to system
      MOV   AX,DGROUP            ; point to data
      MOV   DS,AX
      DISPLAY BANNER             ; print string macro
;
; MLOOP:
      CALL  CD_CONSTAT           ; get status of console device
      CMP   BL,0                 ; bl = 0 means queue empty
      JZ   LOOP2
      CALL  CD_CONIN              ; got one, get the character
      CMP   AL,CC_FS
      JNZ  LOOP1
      RET
;
; LOOP1:
      CALL  CD_AUXOUT             ; got one, send it out
;
; LOOP2:
      CALL  CD_AUXSTAT            ; anything at modem port?
      CMP   BL,0                 ; zero characters in queue?
      JZ   MLOOP
      CALL  CD_AUXIN              ; no, continue main program
      CALL  CD_CONOUT             ; yes, get it
      JMP   MLOOP
      ENDP
      ENDPS
      END
      START
;
; program ends

```

Listing 3. FTYY, the MS-DOS version of TTY for use on the Z100, is written in 8088 assembly language. The support macros in Listings 4 and 5 and the function routines in Listings 6 through 11 are all utilized by this program.

Listing 4. The support macro SEGMACRO.MAC is INCLUDED in FTYY (Listing 3); SEGMACRO contains macros to define the beginnings, ends, and attributes of the segments used in FTYY.

```

; Segment-definition macros for the small-memory model
;
; Data segment
;
DSEG  MACRO  ALIGN
DGROUP GROUP  DATA
DATA  SEGMENT ALIGN PUBLIC 'DATA'
      ASSUME DS:DGROUP
      ENDM
;
; End data segment
;
ENDDS MACRO
DATA  ENDS
      ENDM
;
; Extra segment
;
ESEG  MACRO  ALIGN
EGROUP GROUP  EXTRA
EXTRA SEGMENT ALIGN PUBLIC 'EXTRA'

```



```

ASSUME ES:EGROUP
ENDM

; End extra segment

ENDES MACRO
EXTRA ENDS
ENDM

; Code segment

PSEG MACRO ALIGN
PGROUP GROUP PROG
PROG SEGMENT ALIGN PUBLIC 'PROG'
ASSUME CS:PGROUP
ENDM

; End code segment
ENDPS MACRO
PROG ENDS
ENDM

; Make a stack

SSEG MACRO NUM
STACK SEGMENT PARA STACK 'STACK'
DB NUM&H DUP(?)
STACK ENDS
ENDM

```

FREE		52 PAGE CATALOG AND APPLICATIONS GUIDE	CP/M MSDOS	TRSDOS PCDOS
PP — Plotter Driver Program	For multi and single pen plotters	\$72.95		
XFER — Transfer Function Synthesis	Transfer functions/ Electronic circuits	\$72.95		
TEKCALC — Scientific Calculator Program	Screen graphics/ Statistics/ Programmable	\$72.95		
COMCALC — Communications Design Spreadsheet	Communications budget calculator	\$72.95		
RIGHTWRITER — Report Proofreader Program	Version 2 — Applies 2,200 rules of English	\$97.95		
ACNAP — AC Network Analysis			\$72.95	
DCNAP — DC Network Analysis			\$72.95	
SPP — Signal Processing Program			\$72.95	
PLOTPRO — Scientific Graph Printing			\$72.95	
PCPLOT — High Resolution Graphics			\$72.95	
LOCIPRO — Root Locus Analysis			\$72.95	
ACTRL — Active Filter Design			\$72.95	
STAP — Static Thermal Analysis			\$72.95	
MATRIC MAGIC — Matrix Manipulation			\$72.95	



(714) 781-0252

2200 Business Way, Suite 207, Riverside, CA 92501 U.S.A.

Circle #155 on Reader Service Card

```

; Display '$' terminated string on console

DISPLAY MACRO STRING
  MOV DX, OFFSET STRING
  MOV AH, 09H
  INT 21H
ENDM

```

Listing 5. The support macro DOSMACRO.MAC is INCLUDED in FTTY (Listing 3) to display on the console a string terminated by a "\$".

```

TITLE CDAUXIN - ZENITH BIOS_AUXFUNCTION AUXIN
NAME CDAUXIN

.XLIST
INCLUDE SEGMACRO.MAC
INCLUDE DEFCHR.ASM
INCLUDE DEFBIOS.ASM
.LIST

PSEG BYTE
PUBLIC CD_AUXIN
CD_AUXIN PROC NEAR
  MOV AH, CHR_READ
  CALL BIOS_AUXFUNC
  ; read a char from the aux port
  RET
CD_AUXIN ENDP
ENDPS
END

```

Listing 6. The function CDAUXIN is called by FTTY (Listing 3) to read a character from the auxiliary port.

BIOS_PRNFUNC, BIOS_AUXFUNC, and BIOS_CONFUNC. For their respective devices, these BIOS_XXXFUNC calls will let you read or write a character, get status, etc.

Zenith's version of the MS-DOS BIOS provides these functions in addition to the "plain-vanilla" MS-DOS functions. I have found that sometimes even experienced programmers are unaware of these functions' existence, largely because they were weakly documented when Z-DOS first appeared. At that time, the only discussion of their existence consisted of some INCLUDE files (such as DEFCHR.ASM) and some minimal

comments in Appendix I of the Z-DOS manuals. Now, with the advent of the Programmer's Utility Pack, they are very nicely explained. (See chapter 8 of the PUP manual.)

The listings

Listing 3 gives the Z100 version of the basic TTY program. Listings 4 and 5 provide some support macros to help with segmentation and system calls.

I will also show you how to create a library of routines to call the macros. In this way, you code them only once, and you have them for future use. Listings 6 through 11 give the function routines the

EXAM BUILDER

The Most Comprehensive Exam Storage and Generation System on the Market—In Use In Over 200 Major Universities

- Multiple User-Chosen Subjects
- Complete Question Entry/Edit Capability
- Random or teacher-selected items for finished exam
- Single or Multi-Subject Exams
- Printed copy is ready for reproduction
- Master list print capability
- Password protected answer keys
- Zenith, Apple, TRS-80, IBM

*School Purchase Orders Accepted
No Time Limit Guarantee*

\$99.95

(316) 682-0156

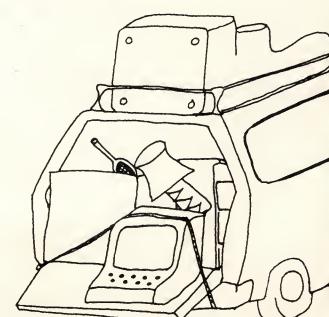
A.U. SOFTWARE—DEPT. ST
P.O. Box 8369
Wichita, KS 67208

Circle #130 on Reader Service Card

MOVING? Don't lose your next issue of *Sextant* in the shuffle.

Send your Change of Address to:

Sextant
Attn: Circulation Department
716 E Street, S.E.
Washington, DC 20003



Index to Advertisers

Reader Service #	Page #	Reader Service #	Page #	Reader Service #	Page #
Analytical Products		Hersey Micro Consulting		Ross Custom Electronics	
* H/Z89 enhancements.	32	168 Console utility for H/Z150-200.	59	160 EPROM programmer.	64
* Word processing software.	64	Hogware		S & K Technology	
Aspen Systems		144 Z100 graphics program.	14	230 Word processor, spelling checkers.	4
149 Subroutine package, editor.	27	ICS Electronics Corporation		Secured Computer Systems	
A.U. Software		115 IEEE 488 Bus controller.	77	157 H/Z89-90 hardware enhancements.	40
130 Examination software.	61	Jay Gold Software		Sextant	
Buss		120 Home finance system.	28	* Subscriptions.	15
* Heath/Zenith news.	31	KEA Systems		* Fifth anniversary offer. back cover	
BV Engineering		175 Communications software; Z100 memory expansion.	73	Sextant Publishing Company	
155 Engineering software.	61	Kandueazy Computer Software Services		246 Books on MS-DOS/Z-DOS and CP/M.	46-47
C.D.R. Systems		119 Accessories & supplies.	51	* Binder interface kits.	63
108 Hardware for H/Z89, H/Z100.	36	Lightek		Sirius Software	
CompuMagic		177 ICs & hardware.	43	212 Hardware, games.	72
156 CP/M utilities.	45	Lindley Systems		Skycastle Computer Products	
Computer Graphics Center		118 Supplies, disks, utilities.	9	218 Printer enhancement software.	73
140 Graphics software for H/Z100-150.	27	118 H/Z100 hard drives.	74	Software Applications of Wichita	
Custom Computer Systems		Matcomm		190 Tax preparation templates for 1-2-3.	28
166 Winchester disk systems.	23	192 Supplies, disks.	8	Software Wizardry	
Cygnus		Micronics Technology		204 Speed enhancement for H/Z151-161.	75
137 Solar system simulation.	40	208 Payroll and printer software.	53	Spectre Technologies, Inc.	
ETTS		207 Hardware for H/Z89, '150, '160.	76	198 CP/M & MS-DOS software.	12
148 Graphics software for H/Z100.	45	Micro Systems Company		UCI Corporation	
FBE Research Company		101 Communications software.	72	180 IBM-PC emulator board for H/Z100.	24
142 H/Z100 enhancements.	5	Payload Computer Systems		196 Hardware for Z100 and Z148.	48
102 H/Z150 enhancements.	8	147 Hardware, software, supplies.	10-11	Barry Watzman	
116 H/Z89 enhancements.	68	P.M.I.		205 Desktop utility software for H/Z100 and H/Z150.	63
First Capitol Computer		250 Z150 hardware enhancement.	16	Workman & Associates	
206 Computer systems.	19	RAM Technology		229 CP/M, MS-DOS software.	27
H & H Computer Enterprises		128 Memory expansion for Z138, '148, '150, 160, '170.	64		
123 Hardware, software for '89; peripherals.	29	Reichert Digital Systems			
Paul F. Herman		151 Accounting software for Z150.	64		
172 Video image capture system.	76				
* Contact advertiser directly.					

Please refer to the key at the right when completing the article evaluation form on the attached Reader Service Card above. By letting us know how interesting and/or helpful you found each feature in this issue of *Sextant*, you will help us choose the best articles for you in the future.

1. Edward A. Byrnes, **Winning Fractals**, p. 6.
2. Wayne Rash, Jr., **Zenith: Planning for the Future**, p. 13.
3. William N. Locke, **Z100 Notebook**, p. 17.
4. William M. Adney, **The Programmer's Utility Pack: Useful Tools for the '100 and '150**, p. 25.
5. Walter J. Janowski, **The Eight-Bit World**, p. 33.
6. William Clarkson, **Four Upgrades for the '89**, p. 37.
7. Joseph Katz, **C Notes**, p. 49.
8. William S. Hall, **Writing Hardware-Independent Terminal Emulators for CP/M and MS-DOS**, p. 56.
9. Victoria Saxon, **CHUGCON 86: Old Friends and New**, p. 70.

Name _____
Address _____
City _____
State _____ ZIP _____

For further information from specific advertisers,
refer to index on page 62 and circle number(s) below.

100 101 102 103 104 105 106 107 108 109
110 111 112 113 114 115 116 117 118 119
120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 139
140 141 142 143 144 145 146 147 148 149
150 151 152 153 154 155 156 157 158 159
160 161 162 163 164 165 166 167 168 169
170 171 172 173 174 175 176 177 178 179
180 181 182 183 184 185 186 187 188 189
190 191 192 193 194 195 196 197 198 199
200 201 202 203 204 205 206 207 208 209
210 211 212 213 214 215 216 217 218 219
220 221 222 223 224 225 226 227 228 229
230 231 232 233 234 235 236 237 238 239
240 241 242 243 244 245 246 247 248 249
250 251 252 253 254 255 256 257 258 259
260 261 262 263 264 265 266 267 268 269
270 271 272 273 274 275 276 277 278 279
280 281 282 283 284 285 286 287 288 289
290 291 292 293 294 295 296 297 298 299

Reader Service Card

January-February 1987

Expires April 15, 1987

Please check *only* one box for each of the following questions.

Which microcomputer system do you own/use?

A. <input type="checkbox"/> H88	E. <input type="checkbox"/> H/Z100-110-120	I. <input type="checkbox"/> Z171
B. <input type="checkbox"/> H/Z89-90	F. <input type="checkbox"/> H/Z150-158-160	J. <input type="checkbox"/> Z200
C. <input type="checkbox"/> H8	G. <input type="checkbox"/> Z138	K. <input type="checkbox"/> Z241
D. <input type="checkbox"/> H11	H. <input type="checkbox"/> Z148	L. <input type="checkbox"/> Z248

How did you obtain this copy of *Sextant*?

1. <input type="checkbox"/> Subscription	3. <input type="checkbox"/> Friend	5. <input type="checkbox"/> Other
2. <input type="checkbox"/> Retail Store	4. <input type="checkbox"/> Single Copy Mail Order	

Please rate the articles in this issue for interest and helpfulness, using the key on page 62.

1 2 3 4 5 6 7 8 9

<input type="checkbox"/>	Excellent
<input type="checkbox"/>	Good
<input type="checkbox"/>	Fair
<input type="checkbox"/>	Poor
<input type="checkbox"/>	Didn't read

Comments _____

62a1

Name _____
Address _____
City _____
State _____ ZIP _____

For further information from specific advertisers,
refer to index on page 62 and circle number(s) below.

100 101 102 103 104 105 106 107 108 109
110 111 112 113 114 115 116 117 118 119
120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 139
140 141 142 143 144 145 146 147 148 149
150 151 152 153 154 155 156 157 158 159
160 161 162 163 164 165 166 167 168 169
170 171 172 173 174 175 176 177 178 179
180 181 182 183 184 185 186 187 188 189
190 191 192 193 194 195 196 197 198 199
200 201 202 203 204 205 206 207 208 209
210 211 212 213 214 215 216 217 218 219
220 221 222 223 224 225 226 227 228 229
230 231 232 233 234 235 236 237 238 239
240 241 242 243 244 245 246 247 248 249
250 251 252 253 254 255 256 257 258 259
260 261 262 263 264 265 266 267 268 269
270 271 272 273 274 275 276 277 278 279
280 281 282 283 284 285 286 287 288 289
290 291 292 293 294 295 296 297 298 299

Reader Service Card

January-February 1987

Expires April 15, 1987

Please check *only* one box for each of the following questions.

Which microcomputer system do you own/use?

A. <input type="checkbox"/> H88	E. <input type="checkbox"/> H/Z100-110-120	I. <input type="checkbox"/> Z171
B. <input type="checkbox"/> H/Z89-90	F. <input type="checkbox"/> H/Z150-158-160	J. <input type="checkbox"/> Z200
C. <input type="checkbox"/> H8	G. <input type="checkbox"/> Z138	K. <input type="checkbox"/> Z241
D. <input type="checkbox"/> H11	H. <input type="checkbox"/> Z148	L. <input type="checkbox"/> Z248

How did you obtain this copy of *Sextant*?

1. <input type="checkbox"/> Subscription	3. <input type="checkbox"/> Friend	5. <input type="checkbox"/> Other
2. <input type="checkbox"/> Retail Store	4. <input type="checkbox"/> Single Copy Mail Order	

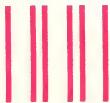
Please rate the articles in this issue for interest and helpfulness, using the key on page 62.

1 2 3 4 5 6 7 8 9

<input type="checkbox"/>	Excellent
<input type="checkbox"/>	Good
<input type="checkbox"/>	Fair
<input type="checkbox"/>	Poor
<input type="checkbox"/>	Didn't read

Comments _____

62a2



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

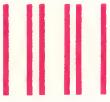


BUSINESS REPLY MAIL

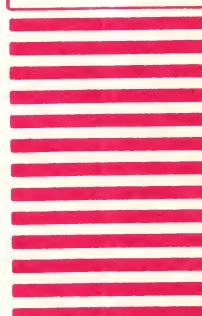
FIRST CLASS PERMIT NO. 27346 PHILADELPHIA, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

**Sextant Reader Service
P.O. Box 13558
Philadelphia, PA 19101**



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

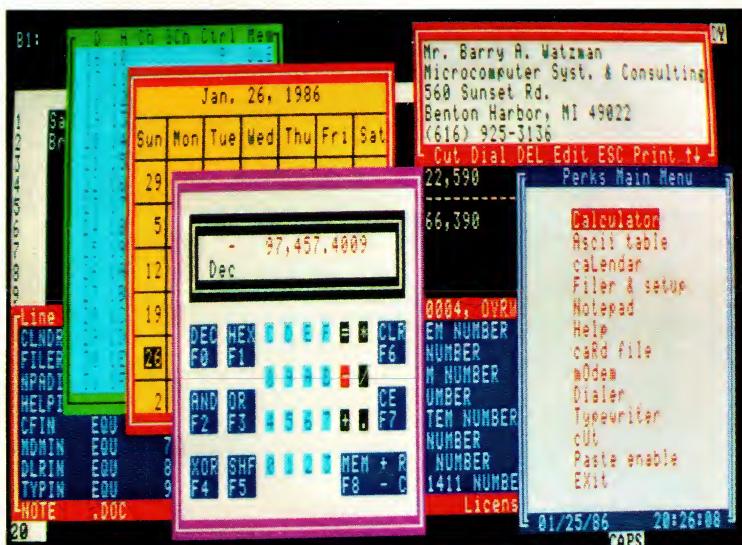
FIRST CLASS PERMIT NO. 27346 PHILADELPHIA, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

**Sextant Reader Service
P.O. Box 13558
Philadelphia, PA 19101**

Announcing Perks™ -PC

Now the Perks desktop utility is available for both the Z-100 series and the Z-100 PC series computers as well as compatible computer systems. Perks-PC includes all Perks version 2 features with command and screen formats identical to Z-100 Perks version 2.



Shown above is an actual photo of Perks in operation, with Lotus 1-2-3 in the background. The main menu and windows for the notepad, calculator, ASCII table, calendar and card file modules are visible.

- Calculator
- ASCII Table
- Notepad w/Variable size buffer (to 64K)
- Perpetual Calendar
- Appointment Calendar
- Context Sensitive Help

- Cut/Paste
- Dialer
- Typewriter
- DOS Functions
- Alarm timers (8)
- Screen Saver
- Import/Export

- Removable w/o rebooting
- Runs with MS-DOS versions 1, 2 & 3
- Modem w/XMODEM file Xfer
- NOT copy protected
- Multi-Key ISAM Card File
- Small Size (under 80K)

Both Perks and Perks-PC are priced at \$69.95 each.* Telephone credit card orders (VISA and MasterCard) are now accepted without a prior signed license agreement.

BARRY A. WATZMAN Microcomputer Systems & Consulting
560 Sunset Road • Benton Harbor, Michigan 49022-7142 • (616) 925-3136

Circle #205 on Reader Service Card

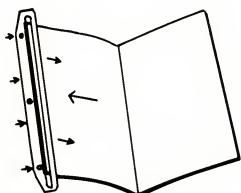
Sextant Binder Interface Kits

Now you can interface your copies of Sextant to a 3-ring binder! Tough plastic Magazine Holders let you preserve, protect, and organize your copies of Sextant without drilling holes in them.

Each package of 12 holders costs only \$2.95. To order yours, return the coupon with your payment today to Sextant Publishing Co., 716 E Street, S.E., Washington, DC 20003.

Instructions

1. Open magazine to center.
2. Slip pages through slot in magazine holder.
3. Slide holder to center of magazine.
4. Place magazine in any standard 3-ring binder.



I want 3-ring binder "holes" for my copies of *Sextant*!

Send me _____ packages of 12 magazine holders \$2.95 each. Please add \$.75 to your total order for shipping and handling.

Payment is enclosed. (Checks must be in U.S. dollars payable on a U.S. bank.)
 Charge my Visa MasterCard

Card # _____ Expires _____

Name _____

Address _____

Send your order to Sextant Publishing Co., 716 E Street, S.E., Washington, DC 20003

MAXIMUM MEMORY FOR YOUR *ZENITH* PC

Z-MAX 150 640/704K upgrade for H/Z 150/160 computers; allows use of one or two banks 256K DRAM. EGA compatible.

\$24.95

Z-MEG 150 1.2 MB upgrade for H/Z 150/160 computers; allows use of up to five banks 256K DRAM. EGA compatible version available.

\$49.95

Z-MAX 138 704K upgrade
Z-MAX 148 704K upgrade
Z-MAX 158 704K upgrade
Permits access of maximum addressable memory.

\$24.95
each

Z-MEG 170 1.0 MB upgrade for Z-171; allows maximum use of up to four banks 256K DRAM.

CALL

No soldering required. All modifications completely reversible. All kits include complete installation instructions, comprehensive trouble-shooting guide, standard 90-day warranty, and RAM-DSK software.

DEALER/DISTRIBUTOR INQUIRIES INVITED



RAM Technology

427-3 AMHERST STREET, SUITE 265
NASHUA, NEW HAMPSHIRE 03063



1 800 662-0070



In Alaska and New Hampshire: (603) 889-0633
HOURS OF OPERATION:
Monday-Friday, 9:00 A.M. - 5:00 P.M. EST

Circle #128 on Reader Service Card

INTELLIBURNER EPROM-EEPROM-MICROCOMPUTER PROGRAMMER

UNIVERSAL PROGRAMMING CAPABILITIES AT AN AFFORDABLE PRICE

- Ultra Fast Programming - 2716's in 16 Seconds
- Programs & Verifies 8K thru 256K Single Voltage EPROMs
- Erases, Programs & Verifies 2815, 2816 & 2817 EEPROMs
- Programs & Verifies 8748 and 8751 Series MICROCOMPUTERS*
- Programming Characteristics Selected by Convenient Personality Jumper Plug (DIP Header)
- Program, Verify, Status, & Diagnostic Display with Tricolor LED
- Serial Interface - 3, 4, or 5 wire - 1200 to 19200 Baud
- Supports ACK/NAK, XON/XOFF and READY/BUSY Protocols

NO SPECIAL SOFTWARE REQUIRED. Transfer disk files (Intel Hex or Motorola's S'Format) to EPROM with your system's line printer or modem software. Transfer EPROM contents to disk file with your system's modem software. Or use the supplied software ** to transfer any binary or ASCII file to/from EPROM.

PROGRAMS:

**NEW EXPANDED FIRMWARE
FOR 27512 & 27513 EPROMs
AND 2864 EEPROMs**

*RS-232C to TTL adapter

SOFTWARE AVAILABLE FOR: **

CP/M systems on 8" SSDS - many 514 formats

TRS-80 Model I & III TRSDOS - NEW! TRSDOS X

Heath H8/H9P HDOS & CP/M

ZENITH 290 & Z100 CP/M - Z100 ZDOS - Z150

KAYPRO II/IV/10/16/286/2000

IBM PC, PC/XT, AT - many MS-DOS systems

IntelliBurner Programmer with Software

Stand-Alone RAM OPTION with 64K Bit RAM

(Expandable to 256K Bit)

RS-232 Interconnect Cable

IntelliBurner PC Boards, EPROM, Plans & Software

\$ 299.00

89.00

12.00

99.00

39.00

Dumburner II Programmer for 28 Pin and

24 Pin EPROMs and EEPROMs (with Software)

Dumburner II Bare PC Board, Plans & Software

16K/32K Dumburner II for 24 Pin EPROMs (with Software)

149.00

39.00

16K/32K Dumburner PC Board, Plans & Software

\$ 199.00

59.00

149.00

39.00

Add \$3 Shipping & handling. (\$2 Bare Board, C.O.D.) is accepted. Foreign Orders add required postage. Specify Environment and Media Requirements.

He, H8, Z100 are "Heath/Zenith; CP/M is "Digital Research. TRS-80 is "Tandy Corp.; IBM-PC, PC/XT, AT are "IBM; MS-DOS is "Microsoft.

Circle #160 on Reader Service Card

NEWWORD - the better word processor by NewStar Software

This fine word processor can be considered a clone of the popular WordStar program and it even uses the same commands. However, NewWord is faster, easy to customize, has the mail merge feature built-in, supports more printers, has an UNDELETE function and excellent documentation. It also includes the WORD Plus spelling checker by Oasis — an excellent product in itself! version 3 includes Macros and other enhancements. We offer this package at a discount. Just check the prices below.

Available in most computer formats including Heath hard sector and BIOS-80.

30 Day Money-Back Guarantee!

CP/M 80 \$145
MSDOS or CP/M 86 \$195
MSDOS version 3 \$275

Add \$3 per order for shipping and handling.

Terms: Check or Money Order — VISA/MC — COD. California residents add 6% tax.

ANALYTICAL PRODUCTS

805/688-0826

213 Teri Sue Lane

Buellton, CA 93427

**NEW
RELEASE 1.6**

CPA II

Personal Accounting
for IBM PC Compatible Systems

A popular system now made even better! All the features of our earlier system, with many new ones. But, best of all...

NEW Pricing... Now only \$59.00

NEW Features and Capabilities

NEW Hierarchical Menus

NEW Easy Set-up Procedure

NEW On-line Help Screens

NEW Mortgage, Loan, Investment and Depreciation Utility

Any number of Vendor, Cash and Checking accounts desired. User-specified Expense and Income categories. Check printing, auto-posting, spreadsheets, bar graphs, tax deductions, accounts-coming-due, budgets, and many other detailed reports. Auto-checkbook reconciliation. Easy user-command entry. Full edit capabilities. Fast and efficient. Can be used with any printer for all reports and bar graphs. Not copy-protected.

Requires 256K RAM, color video, RGB monitor, at least 2 disk drives (2 floppies, or 1 hard disk and 1 floppy), MS-DOS 2.1 or greater. Monochrome version to be announced.



REICHERT DIGITAL SYSTEMS
29 BLAZIER RD. WARREN, N.J. 07060 (201) 469-3854

Circle #151 on Reader Service Card

program will call.

(Also, the program relies upon some standard Z-DOS support files which may not be available to those of you using MS-DOS 2 without the Programmer's Utility Pack. Below and in Listings 12 and 13, I give some substitute code.)

The background

Recall from Part I that our biggest problem is to check the status of the console and communication (or auxiliary) devices.

Standard MS-DOS offers function calls that are rather similar in purpose to

CP/M's and have some of the same limitations. In particular, there is no way to check the status of the communications port. In CP/M, dynamically changing the IOBYTE can remedy this situation. In MS-DOS, there is nothing quite like the IOBYTE. (There is a way to redirect standard input and output in version 2 or greater. However, I shall not cover that in this article.)

As noted, however, Zenith has augmented the standard system calls with an additional group to allow the Z100 to access the character devices.

As I mentioned above, the auxiliary ports on the Z100 are buffered; they are also interrupt-driven. This means that when a character appears at the port, the communications chip sends an interrupt signal to the CPU; the CPU then completes its current instruction, saves the environment, gets the character, puts it into a buffer structure called a FIFO (first in, first out) queue, and returns to whatever it was doing. This operation is transparent to the user.

Being buffered and interrupt-driven also means that the port can handle fairly high input rates without losing characters. Instead of checking the status of the hardware device at the port, it is enough to see if anything is in the buffer, which on the Z100 is 255 characters long. With the BIOS_XXXFUNC calls, the Z100 version of MS-DOS provides a means for checking this queue.

All other functions needed by the TTY program (namely, check keyboard status and read the keyboard, write to the screen, and read from and write to the communications port) can be done with standard MS-DOS system calls similar to those in CP/M. When I upgraded my machine to MS-DOS 2, however, I found that the standard system calls were simply too slow. My Z100 had certainly been fast enough under Z-DOS, even when the communications port was running at 4800 baud. MS-DOS 2, however, just has more to do and more to check up on.

Because of MS-DOS's slow speed, I decided to do *all* I/O operations using the Z100 version's character-device routines. This is not quite in the spirit of trying to use the standard operating system for all services and remaining machine independent; nonetheless, we will still be operating a level above direct manipulation of the hardware.

Unfortunately, one must still get over the barrier of writing something in 8088 assembler.

I could have written the Z-DOS program in a .COM version, which would have followed the 8088's "8080 model." A .COM program would be much like the CP/M version. And it would be subject to the same limitations as any CP/M program—mainly an inability to work in more than 64 kilobytes of memory (one segment of memory).

```
TITLE CDAUXSTAT - BIOS_AUXFUNC AUX STATUS
NAME CDAUXSTAT

.XLIST
INCLUDE SEGMACRO.MAC
INCLUDE DEFCHR.ASM
INCLUDE DEFBIOS.ASM
.LIST

PSEG BYTE
PUBLIC CD_AUXSTAT
CD_AUXSTAT PROC NEAR
    MOV AH,CHR_STATUS
    MOV AL,CHR_SFGS
    CALL BIOS_AUXFUNC
    RET
CD_AUXSTAT ENDP
ENDPS
END
```

Listing 7. The function CDAUXSTAT is called by FTTY (Listing 3) to get the length of the queue at the auxiliary port.

```
TITLE CDAUXOUT - BIOS_AUXFUNCTION AUXOUT
NAME CDAUXOUT

.XLIST
INCLUDE SEGMACRO.MAC
INCLUDE DEFCHR.ASM
INCLUDE DEFBIOS.ASM
.LIST

PSEG BYTE
PUBLIC CD_AUXOUT
CD_AUXOUT PROC NEAR
    MOV AH,CHR_WRITE
    CALL BIOS_AUXFUNC
    RET
CD_AUXOUT ENDP
ENDPS
END
```

Listing 8. The function CDAUXOUT is called by FTTY (Listing 3) to write a character to the auxiliary port.

```
TITLE CDCONIN - ZENITH BIOS_CONFUNCT CONIN
NAME CDCONIN

.XLIST
INCLUDE SEGMACRO.MAC
INCLUDE DEFCHR.ASM
INCLUDE DEFBIOS.ASM
.LIST

PSEG BYTE
PUBLIC CD_CONIN
CD_CONIN PROC NEAR
    MOV AH,CHR_READ
    CALL BIOS_CONFUNC
    RET
CD_CONIN ENDP
ENDPS
END
```

Listing 9. The function CDCONIN is called by FTTY (Listing 3) to read a character from the console port.

Instead, I elected to write an .EXE version, which means that it is written in the form proper to the 8088: it uses segmentation, the 8088's method of accessing memory larger than 64K. In the program given here, I use only one stack segment, one data segment, and one code segment; such an arrangement is called the small-memory model.

An .EXE program is more difficult to write than a .COM version, but in the end it is the best way to learn and exploit the complexities of the assembler and the benefits of segmentation. (As multi-user/multi-tasking systems such as Windows and Xenix begin to appear, it becomes increasingly necessary to learn to create .EXE programs. First, they are relocatable, and multi-tasking requires that programs be loadable anywhere in memory—not just at some fixed point, as with a .COM program. Also, the memory requirements of many of these programs make segmentation a must.)

(If you want some more information on writing programs using the small-memory model, you might take a look at my article in *Sextant* #17, July-August 1985, "OFFLIN: An Exercise in Z100 Assembly Language.")

The program

Let's read through the program in Listing 3. It begins with a TITLE statement, followed by several comments about the program. Then, a standard Z-DOS support file is INCLUDED: DEFASCII.ASM, a file giving values to control characters.

The next two INCLUDE files are macros that I have written, and they are found in Listings 4 and 5. A macro is simply text which is assigned a name; typically, it is a set of program lines. When a macro's name is used in a program, the text is substituted for the name.

SEGMACRO.MAC, in Listing 4, assists in arranging segments. DOSMACRO.MAC, in Listing 5, contains but one macro. That macro prints any string that is terminated by a "\$". Similar macros can be written to include other MS-DOS function calls.

DEFASCII contains EQUATES (symbolic names) for the non-printing ASCII characters. Ordinary ASCII text files will need just the carriage return (hex value, 13) and line feed (10h); and we'll use the CTRL-\(28h) as our exit character. So, if you do not have a copy of DEFASCII.ASM, you can substitute the following lines in place of INCLUDE DEFASCII.

```
CC_CR EQU 13
CC_LF EQU 10
CC_FS EQU 28
```

The file SEGMACRO.MAC in Listing 4 is just a collection of macros. Basically, SEGMACRO's macros take a lot of assembler and linker housekeeping and put it in one convenient place.

For instance, we must establish the alignment of segments; that is, whether a segment's boundary can be on any arbitrary byte, say, or on a paragraph (low-

```
TITLE CDCONSTAT - BIOS_CONFUNCT CON STATUS
NAME CDCONSTAT

.XLIST
INCLUDE SEGMACRO.MAC
INCLUDE DEFCHR.ASM
INCLUDE DEFBIOS.ASM
.LIST

PSEG BYTE
PUBLIC CD_CONSTAT
CD_CONSTAT PROC NEAR
MOV AH,CHR_STATUS
MOV AL,CHR_SFGS
CALL BIOS_CONFUNC
RET
CD_CONSTAT ENDP
ENDPS
END
```

Listing 10. The function CDCONSTAT is called by FTTY (Listing 3) to get the length of the queue at the console port.

```
TITLE CDCONOUT - BIOS_CONFUNCT CONOUT
NAME CDCONOUT

.XLIST
INCLUDE SEGMACRO.MAC
INCLUDE DEFCHR.ASM
INCLUDE DEFBIOS.ASM
.LIST

PSEG BYTE
PUBLIC CD_CONOUT
CD_CONOUT PROC NEAR
MOV AH,CHR_WRITE
CALL BIOS_CONFUNC
RET
CD_CONOUT ENDP
ENDPS
END
```

Listing 11. The function CDCONOUT is called by FTTY (Listing 3) to write a character to the console port.

order four bits of the address are zero). Also, the assembler needs ASSUME statements in order to know in which segments the variables are contained.

As well, we have to give a group name to segments in order to tell LINK which segments should be loaded together. Using the GROUP directive ensures that in the final program there is only one code, one data, one stack, and (if needed) one extra segment. The class parameter of the SEGMENT directive will group segments under a name in order to handle the actual order in which segments are loaded by LINK.

I have found that by using such macros, people are much less intimidated when trying to write .EXE programs. The main program becomes easier to write (and read), and you tend to make far fewer errors. In addition, macros make it much easier to code the main program so that it can access library functions (as I plan to do).

After the three INCLUDE statements in Listing 3, a stack is established by invoking the SSEG macro with the parameter 100 to define a stack segment with 256 (100 hex) bytes of space. A stack is used

for temporary storage, such as holding the values needed by subroutine calls. It is an essential data structure in most programs.

In an MS-DOS program, however, it is not always necessary to actually define an explicit stack segment since the default stack may suffice. .COM programs, for example, reside entirely in one segment, the code segment. So, they cannot have a separate stack segment, although a local stack may certainly be created by the programmer within the code segment.

.EXE programs written as device drivers or resident programs may or may not need a stack since the operating system or currently running program can provide the service. But stand-alone .EXE programs generally require a separate stack segment.

Next, the DSEG macro establishes a data segment which will have a paragraph alignment. As noted above, it will start on a paragraph boundary: an address whose low-order four bits are zero. This alignment is accomplished with the PARA parameter. (The group name, class name, etc., are also set by the macro; these are hidden in the text of the macro itself.)

```

; CHRDEV.H

CHR_WRITE    EQU    0           ; write function
CHR_READ     EQU    CHR_WRITE+1  ; read function
CHR_STATUS   EQU    CHR_READ+1  ; status function
CHR_SFGS     EQU    0           ; get status function
CHR_SFGC     EQU    CHR_SFGS+1  ; get configuration information
CHR_CONTROL  EQU    CHR_STATUS+1 ; control function
CHR_CFSU     EQU    0           ; set up new function
CHR_CFCI     EQU    CHR_CFSU+1  ; clear input subfunction
CHR_CFCO     EQU    CHR_CFCI+1  ; clear output subfunction
CHR_LOOK     EQU    CHR_CONTROL+1 ; non-destructive read function
CHR_FMAX    EQU    CHR_LOOK    ; maximum function number

```

Listing 12. FTTY requires DEFCHR.ASM to be present during assembly. If you do not have access to DEFCHR.ASM (from either Z-DOS or the Programmer's Utility Pack for MS-DOS), CHRDEV.H supplies you with the necessary code to replace it.

```

; BIOSSEG.H

BIOS_SEG SEGMENT AT 40H

ORG    24*3      BIOS_DSKFUNC  LABEL  FAR      ; disk function
ORG    25*3      BIOS_PRNFUNC  LABEL  FAR      ; printer function
ORG    26*3      BIOS_AUXFUNC  LABEL  FAR      ; aux function
ORG    27*3      BIOS_CONFUNC  LABEL  FAR      ; console function
BIOS_SEG ENDS

```

Listing 13. FTTY requires DEFBIOS.ASM to be present during assembly. If you do not have access to DEFBIOS.ASM (from either Z-DOS or the Programmer's Utility Pack for MS-DOS), BIOSSEG.H gives you the necessary code to replace it.

Generally, in the main module of an .EXE program, segments are given paragraph alignment. Other segments (which may be defined in other modules) in the same group are then aligned on BYTE boundaries. Thereby, when the program is finally linked, the various pieces that make up the final segment have no gaps but are contiguous.

In this program, the data segment contains only a banner which is printed when the program begins. Notice the ENDDS macro to end the data segment.

Finally, the PSEG macro creates the code segment, with paragraph alignment.

PSEG is followed by the declaration of a procedure called MAIN. One reason for using procedures is to modularize the code. Also, it saves a bit of coding in that we will not have to worry about coding the proper return; the assembler will do it for us.

Here, MAIN is a FAR procedure; that is, its call and return have to cross between segments. MAIN is called by the operating system when the program is loaded and starts to execute. Since the call originates within the operating system's segment, MAIN must make a far call to get to the program and a far return to get back to the operating system when the program finishes.

The assembler will code the correct

type of return according to the type of procedure in which it is contained. If a procedure is tagged as NEAR, then a near return will be assembled.

It is important to get these procedures coded correctly. Before the jump is made, a far call pushes onto the stack both the code segment address and the logical (offset) address to which it must return. A far return will pull both addresses from the stack and thereby continue to run correctly.

Since it is to remain in the same segment, a near return will alter only the logical (offset) address. If a far return is needed, then the program will obviously be in the wrong segment; but it will, of course, continue to try to execute—with unpredictable results.

Just after the label MAIN, several external routines are declared. These routines reference library routines, which are found in Listings 6 through 11. (Later, I will show you how to use them as a library to be incorporated into the program during the assembly process.)

With those assembler directives out of the way, the actual code begins at START. There, three lines save the current data-segment register onto the stack and then PUSH an offset of 0 onto it.

These three lines ensure that we can make a graceful return to MS-DOS at the end of program operation.

When the program starts executing, MS-DOS hands it the beginning address of the data segment. If the code-segment register points to the beginning of the data segment, an exit to MS-DOS can be made by executing a far return to that address. This is actually the beginning of the program segment prefix, sometimes called the program header. This is prefixed to a program by the operating system in order to provide a variety of information (memory size, etc.).

You can examine the program segment prefix in DEBUG by loading your .EXE program and giving the command D(display)DS:0. At bytes 0 and 1 is CD 20, the exit instruction for a return to the operating system—INT 20h.

If the code segment is set as described above with the instruction pointer set to 0, then interrupt 20 will be executed and a return to the operating system will be accomplished. The far return instruction effectively sets everything correctly. First, it will pop off the stack the 0 we pushed there, then put it into the instruction-pointer register. Then it will pop the value of the data-segment register (which we also pushed onto the stack), and put it into the code-segment register.

(If you occasionally get a bit confused, you may want to accompany this article with a reading of my OFFLIN article in *Sextant* #17. And don't feel bad. It is *not* really easy to keep these things straight, either in writing a program or in reading one.)

Those first three lines after START, then, take care of our eventual return to MS-DOS. With that out of the way, we can get to the actual operation.

The next two instructions in the code establish our own data segment in the group DGROUP, whose exact definition can be found in the segment macros.

The banner is then printed on the screen by using the print-string function call (9). I have incorporated function 9 into DOSMACRO.MAC, which displays the string passed to it. After the banner is displayed, the program begins its main loop.

The main loop follows the general pattern given in the CP/M discussion. First the keyboard queue is checked for a character by calling CD_CONSTAT. (See Listing 10.) Before it returns, this function will load BL (the low byte of the BX register). If BL is 0, then nothing is waiting to be read, and the program jumps to LOOP2.

If BL is non-zero, then the character is read by calling CD_CONIN. (See Listing 9.) The character read is checked to see if it is the exit character. If it is, the program returns to the operating system; if it's not, the program is sent to the communications port or AUX: device using CD_AUXOUT. (See Listing 8.)

The pattern repeats itself, this time using the status and input routines for the AUX: device. The program then returns to

the top of the loop. Except for the check for the exit character, the code for the CON: and the AUX: devices are symmetric to each other.

Building a library of character device calls

As discussed above, the program references the six external routines given in Listings 6 through 11. I have coded them as NEAR since I expect that, in most programs, they can all be placed in the same segment. If you ever change to FAR procedures, then you will also have to change SEGMACRO.MAC to incorporate different segmentation macros.

These six routines all follow the same pattern. None needs a data, stack, or extra segment. The code segment in each is determined by the PSEG macro. Relying on PSEG ensures that only one code segment will exist in the final .EXE program, since PSEG puts into PGROUP all code segments in both the external routines and the main program.

Note that all these routines make far calls to names which are defined in the INCLUDE files DEFCHR.ASM and DEFBIOS.ASM; these files are supplied with the MS-DOS 2 Utility Pack. You will also find similar files with Z-DOS, although the names may be slightly different.

DEFBIOS.ASM defines the BIOS segment and gives the addresses of the various BIOS_XXXFUNC calls that our program

will use.

DEFCHR.ASM gives a complete list of constants, definitions, etc., needed for the various types of operations that can be performed on character devices. These operations include many more than just checking queue status and reading and writing characters.

In case you do not have access to DEFCHR.ASM and DEFBIOS.ASM, I have supplied the necessary pieces of them in the two headers, CHRDEV.H and BIOSSEG.H, given in Listings 12 and 13. (These originally appeared in my article, "Switching Z100 Printer Ports," in the August 1985 issue of REMARK magazine.)

Each of the six routines in Listings 6 through 11 can be placed into an object-code library. If you do any serious work with .ASM, you simply must build libraries rather than code the same procedures over and over.

To accomplish this, use an editor to type the procedure, then assemble it with MASM. If there are assembly errors, fix them and re-assemble.

Now run LIB to produce the library, naming it CHRDEV. When LIB prompts you for an action, enter +FILENAME, where FILENAME is the object file you just generated with the assembler. The listing file can be called CHRDEV.LST. You can repeat the operation to add more files, or you can add several at once with the operation

+ FILE1 + FILE2 + FILE3 + etc.

With TTY.EXE, we won't be using the six routines on their own; so, the .OBJ file produced by assembling the procedure can be deleted after exiting LIB. Save the sources, however. To save them under one name (and to reduce disk space usage) you can use the BACKUP program that comes with Z-DOS's Winchester utilities. (RESTORE will get the files saved by BACKUP back as separate files.) Other utilities, such as Archive from Generic Computer Products, do the same thing.

Later, when you are linking a program and the request for libraries is made, you can use CHRDEV.LIB as one of the inputs. It will be searched, and only those external routines in the library that are needed in your program will be loaded. If you put all six of the external routines I have supplied into a library, then you will have the basis for a very useful set of procedures.

Assembling and running TTY

After typing the program, name it TTY.ASM and run MASM. You will be asked for the input file; answer with TTY. The object file defaults to TTY.OBJ. If you want a listing or cross-reference file, answer appropriately. If not, just enter a return.

If the program assembles correctly, run LINK. The program name is again TTY, and the run file defaults to TTY.EXE. If you want a list of symbols (labels and equates), answer TTY to the question about a MAP file. This file can also be routed to the

console if you type CON instead of TTY.

Finally, answer CHRDEV (or whatever you have called the file of external procedures) to the request for libraries. If you have not yet built the library, then you can simply give the names of the seven object modules required when the linker first asks you for a file name. (See the LINK documentation for more detail.)

To run the program, simply type TTY. The program assumes that you have a modem connected to the auxiliary port and that you have configured the port for the correct baud rate, word size, parity, port address, etc.

You may then attempt to communicate with another computer. Or you can short together the AUX port's data-in and data-out pins (2 and 3) to test your program. (Alternatively, your modem may have an analog-loopback procedure.)

You cannot do much more than talk to another computer, but you have the basis for writing a program to do more, and you have increased your understanding of your machine and its resources.

You can, though, do some crude file transfer. After connecting with the remote computer, exit to the system and give the command COPY FILENAME AUX:. The file will be sent to the other system.

As with the CP/M version of TTY, however, the file may arrive with extra line feeds. The remote end will probably expect you to send only a carriage return, without a line feed; if that's what it expects, it will usually treat the following line feed as another carriage return.

As simple as this program is, I find it adequate for an electronic-mail service such as MCI Mail. I could upload a file with COPY if it weren't for the line-feed problem, but that's taken care of easily enough.

Before calling MCI Mail, I first compose and file the letter to be sent. Then I use the TTY program to call MCI and get into their editor. I then exit to MS-DOS and run a simple C program that does nothing more than eat line feeds while copying the file to the AUX: port. I then run TTY again to talk to MCI and send the letter.

In the days when everyone has Cadilacs for communications software, it is still fun to drive a Model T—especially when you've made it yourself.

Ordering Information

MS-DOS 2 Programmer's Utility Pack, model #CB-5063-16, \$149.
Heath Company
Benton Harbor, MI 49022
800/253-0570; from Michigan and Alaska, 616/982-3411

Archive, \$34.95.
Generic Computer Products, Inc.
P.O. Box 790, Dept. BD86
Marquette, MI 49855
906/249-9801



FBE Cards For H/Z-89 Computers

SPOOLDISK™89 Electronic Disk & Printer Interface/Spooller—Micro-processor-based card with 128K of RAM acts like fast, quiet disk and also runs printer via Centronics Parallel interface (RS-232 available). Use for normal system printing or for spooled automatic printing. Installs on right-side bus. CP/M® and HDOS software included. **RS232 \$15**

SPOOLDISK™89 \$195
Parallel Printer Cable \$24

SLOT4 BUS EXTENDER—Adds fourth card slot to right-side bus. Easy installation with no permanent modifications. **SLOT4 \$47.50**

H89PIP PARALLEL INTERFACE—Two ports using 8255 PPI chip. One port bare and one port buffered for use as Centronics Parallel printer interface. Installs on left or right-side bus. CP/M® and HDOS printer driver software included. **H89PIP \$50**

Parallel Printer Cable \$24



FBE Research Company, Inc.

P.O. Box 68234

Seattle, WA 98168

(206) 246-9815

CP/M is a trademark of Digital Research Inc.
VISA and MasterCard Accepted.

Circle #116 on Reader Service Card

SEXTANT BACK ISSUES

#1 SPRING 1982

A Parallel Interface for the '89
Disk Programming Without HDOS
Print Spoolers

#11 JULY-AUGUST 1984

Review of the Z150
Introduction to Pascal: Part 1
Inexpensive Letter-Quality Printers

#20 JAN.-FEB. 1986

Five Desktop Utilities for the Z100
Index to *Sextant* #1-#20
Customizing muLisp for '89 or '100

#2 SUMMER 1982

Crucial Secrets of HDOS
Preview of the Z100
Structure in BASIC Programming

#12 SEPT.-OCT. 1984

Z-DOS Communications Software
Introduction to Pascal: Part 2
The HUG Bulletin Board

#21 MARCH-APRIL 1986

8087 Math Chip Speeds Up '100
Using PROMPT on the '100 and '150
See Seven Sorts

#3 FALL 1982

The World's Fastest Sort?
Give Your '89 an Automatic Boot
Graphics on Your H8

#13 NOV.-DEC. 1984

A Parallel Port on Your '89
Introduction to Pascal: Part 3
Making Z-BASIC Friendlier

#22 MAY-JUNE 1986

ZCLK for the Z100
Run CP/M under MS-DOS
'100 and '150 Monitor Routines

#4 WINTER 1983

HDOS Interrupt Programming
Eight Spreadsheets Compared
Beyond the H/Z Disk Controller

#14 JAN.-FEB. 1985

Microsoft Word and Mouse
Grafting an 8086 Onto the '89
WatchWord for the Z100

#23 JULY-AUGUST 1986

BASIC Fractals
1216K of RAM for Your '151
A Low-Budget HDOS Data Base

#5 SPRING 1983

CP/M: The First 13 System Calls
Understanding HDOS
Improve Your H8: Add a Fan

#15 MARCH-APRIL 1985

Teach Your Z100 Greek
Graphs on the '100 and '150
HDOS Command Processing

#24 SEPT.-OCT. 1986

The Z150 and Gate Arrays
Subconscious Memory of the Z100
Print Big in BASIC (or Turbo Pascal)

#6 SUMMER 1983

Pascal Elucidated
Assembly Language In a Spooler
Passwords and Record-Keeping

#16 MAY-JUNE 1985

Great Graphics on the '19 and '89
Speed Up Your '100 50%
A Critical Look at the 8088

#25 NOV.-DEC. 1986

Turbo Pascal
An MBASIC Simulation
Upgrade the ZVM-134 to 16 Colors

#7 FALL 1983

Z100 Software Roundup
Make Your '8 Think Like an '89
dBASE II Comes to Heath/Zenith

#17 JULY-AUGUST 1985

Moving Up to MS-DOS 2
Auto Key Repeat for the '89 or '19
Affordable Graphics for the Z100

#8 WINTER 1984

Games Roundup
Z-DOS Assembly Language
Kids and Computers

#18 SEPT.-OCT. 1985

The Ultimate Word Processor?
Menu Control with HDOS MBASIC
'89 Interactive Graphics Controller

#10 MAY-JUNE 1984

Microsoft Windows
Color Graphics for your '8 or '89
Put an Electronic Disk into Your H8

#19 NOV.-DEC. 1985

Speeding Up the Z100
HDOS Enhancements
S-100 Modem Upgrade

**Complete your
Sextant library by
ordering missing
back issues—don't
miss any valuable
information for your
Heath/Zenith
system!**

\$3.50 each

Send me *Sextant* issues:

#1 #11 #20
 #2 #12 #21
 #3 #13 #22
 #4 #14 #23
 #5 #15 #24
 #6 #16 #25
 #7 #17
 #8 #18
 #10 #19

_____ issues @ \$3.50 ea. = \$ _____

(Please include \$1.50 extra for each issue
mailed overseas.)

Payment enclosed (checks must be in
U.S. dollars payable on a U.S. bank)

Charge my
 Visa MasterCard

Card # _____

Exp. date _____

Name _____

Address _____

Clip coupon and send today to:

Sextant
Dept. S26
716 E Street, SE
Washington, DC 20003

CHUGCON 86: Old Friends and New

This year's Capital Heath Users' Group Conference showcased some new products and new vendors. It was also the setting for a donation to the Smithsonian.

Victoria Saxon

Do you know how many computer programmers it takes to screw in a light bulb?

None.

(It's a hardware problem.)

If you don't believe me, just ask Sol Libes. He was the banquet speaker for CHUGCON 86, the fifth annual Capital Heath Users' Group Conference. It was held during the weekend of October 25-26, 1986, at the Sheraton Hotel in Tysons Corner, Virginia, just outside Washington, D.C.

Libes is currently the editor of *Micro/Systems* magazine, and has written for *Byte*, *PC World*, and many other computer magazines. For CHUGCON attendees, however, he provided a special 45 minutes filled with rumors, anecdotes, and computer jokes; they had, he said, been deleted from past columns for one reason or another by his editors.

Since our magazine is rated for general audiences, we can't tell you everything that Libes told us, but we can tell you some of it.

Take this one, for instance: There's a rumor going around that software publishers are going to start selling advertising space in their programs in order to earn a little extra money.

Can you imagine, asked Libes, waiting for your system to calculate a spreadsheet—then suddenly, "Aren't you hungry for Burger King now?" pops up on your screen? Or how about this: "If this were a Compaq DeskPro 386, you'd be finished now."

Libes also was gracious enough to tell us the difference between hardware and software. Hardware: if you play with it long enough, it breaks; software: if you play with it long enough, it works.

Victoria Saxon is a former Sextant employee. She now works for the U.S. Senate, but still likes to go to HUG conferences. She wrote this article on her own Z120.

Photo by Charles Floto



The local Heath/Zenith Computers and Electronics stores decided to sell keyboards for a dollar a piece on the last afternoon of the conference. A lot of people took them up on the deal.

New faces

P.C. Enterprises made its debut at the conference with a brand-new accounting program called *Annuity*. P.C. Enterprises' management consultant, Edythe Frankel, and systems consultant, Gary Kelleher, seemed very excited about becoming new members of the Heath/Zenith community.

In fact, Frankel said she was inspired to write a program for the H/Z89 by her father; he owns an '89, and told her she might find an enthusiastic software market among '89 users. (The '89 version of *Annuity* runs under CP/M and is available on hard- or soft-sector disks.)

Frankel didn't stop with just the '89, though. *Annuity* also runs on the Z150 and other IBM clones. And during the second day of the conference, Frankel and Kelleher actually figured out a way to make the program run on the Z100, too! (These are not the type of people who sit around doing nothing at computer conferences.)

Annuity is designed for both advanced money managers and "novice accountants." The accounting terms used in the program are explained in a README file on the disk.

Annuity will help you solve many of your financial problems. For example, if you are interested in saving money to buy a car or send a child to college, it will help you figure how much you need to put away every month, how much interest you should earn while you make these deposits, and even how long it's going to take you to reach your goal.

As Frankel said, *Annuity* is helpful for letting you know whether what you are doing "is realistic or not."

It also does many other things, and if you are interested in becoming a financial planner, you may want to invest your first \$19.95 in this program.

Goldstein Software, Inc. was another newcomer to the Heath/Zenith arena, offering new programs for the IBM compatibles. For those of you who do

technical writing, their program "Tech/Word" may be just what you have always wanted. Priced at \$350, Tech/Word helps you use any IBM-compatible computer to display and print technical symbols in your documents.

For example, the program provides automatic screen formatting for mathematical expressions; this makes it easier to write fractions, subscripts, superscripts, and other figures not on the normal horizontal line of text. Tech/Word also provides math symbols, such as the integral sign, sigma, and pi.

Tech/Word prints Greek characters, and it also allows you to create and program your own symbols in case the program missed any. According to Goldstein's director of marketing, Jim Cook, the program is also useful for typing foreign languages: you can even re-adjust your entire keyboard to type the Cyrillic or Hebrew alphabet!

Goldstein Software also offers GoldWord, a word processor (priced at \$39.95). Another program, GoldSpread, is a spreadsheet program similar to Lotus 1-2-3, with 2,048 rows by 256 columns. Priced at \$59.95, GoldSpread provides data management and graphics capabilities. When Cook was asked to distinguish his company from its competitors, he said, "We are the least expensive." (Sounds good to me.)

Travis Barfield almost made it to last year's CHUGCON as a vendor, but decided at the last minute that he wasn't ready. This year, however, he showed up with lots of enthusiasm and three new programs. His best-selling program at the conference was "Sextant," a program which he named after a highly rated and extremely popular magazine for Heath/Zenith computer users.

Sextant allows you to track satellites at take-off and in orbit. You can choose from among the different launch sites of the United States, Japan, China, the Soviet Union, and the European Space Agency.

Barfield's demonstration run showed a map of the world with lots of satellites sailing across the screen. Not only that, but Sextant comes in color, too! This program and Barfield's other programs are all for the H/Z100, and run under MS-DOS.

"Printer's Devil," another Barfield program, is designed to let you use your computer as you would use an intelligent typewriter. Barfield says he wrote it after getting confused by the numerous commands provided in most word processing packages.

The third program offered by Barfield is "Geometry." Barfield said he wanted the program to help people see everyday objects as geometric figures. For example, a road could be a rectangle. The program provides a math scratch pad, and can calculate the volume, length, width, and area of various geometric

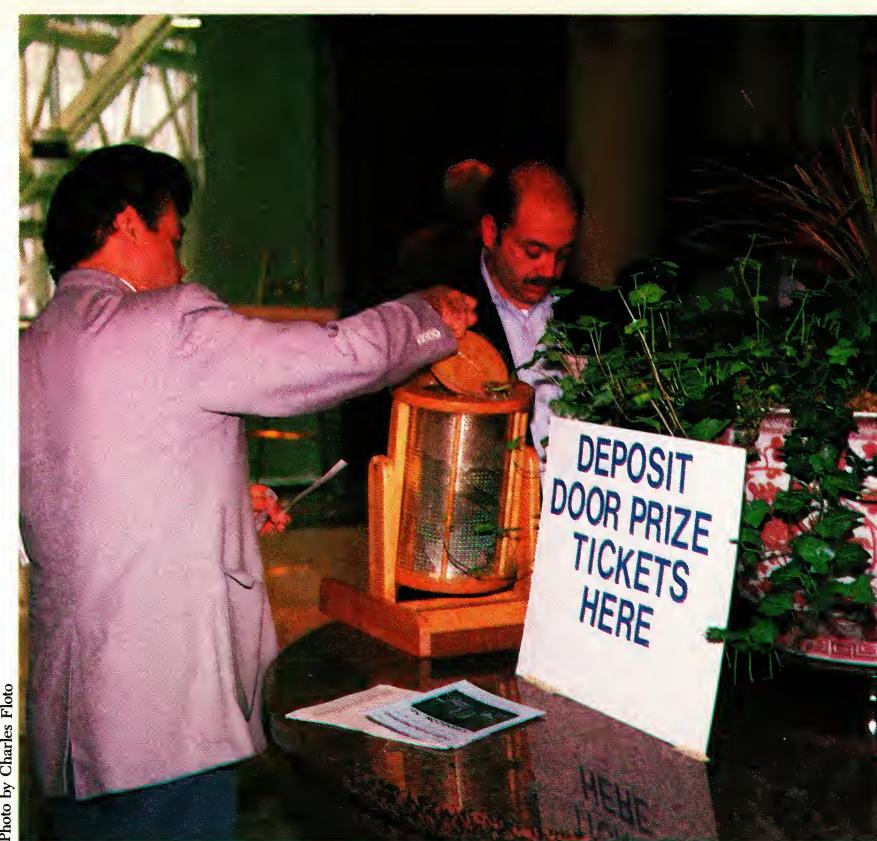


Photo by Charles Flock
Vendors who had booths at CHUGCON 86 had donated door prizes that were raffled off every hour during the conference. If you registered, you were eligible.

objects.

The fourth and final new vendor at CHUGCON was *Quic-n-Easi Products, Inc.* (QNE). QNE specializes in quick and easy management programming aids for business people who do not have the time to program.

Q-Pro 4, one of their products, is designed to help create business applications according to the user's taste. One of the features offered by this package is two programming levels: "fill-in-the-blanks" and the "interpreter". The former allows a user to do such things as create custom screen formats without writing code. The latter allows the programmer to go beyond format building, and manipulate data in various ways. Apparently, it would take a user hours of programming to produce an equivalent program in BASIC.

QNE introduced a new program, QNE2C, at the conference. QNE2C will translate a Q-Pro 4 program into C source code. The advantage to this is that the C version will run much faster.

New ideas

And speaking of "new" things, Cornelius Mhley, CPA, gave a talk entitled "The New Tax Law and You (for Microcomputer Users)."

Mhley had advice for workers who have computer-based businesses "on the

side": Get some outward, visible signs that you really are in business. He suggested keeping a diary of contacts, getting stationery and business cards with your business name on them, printing an inexpensive brochure, and/or getting a separate business phone number in your house. In 1987, this will help you prove (with documentation) that you do indeed run a legitimate business in your home even if you also are employed elsewhere.

Finally, the Capital Heath Users' Group supplied something new of their own: TEDI (Telecommunications Exchange for the Deaf, Inc.) provided signers for the deaf at the conference.

Old friends

There were several familiar faces at CHUGCON. Terry Weber of *Weber & Sons, Inc.*, was handing out free samples of his replaceable labels for disks and disk jackets; and the people at *Gemini Technology Corp.* were busy talking about PEGASys+, their brand-new graphics adapter for the H/Z100 and the IBM compatibles.

Lindley Systems was showing off an absolutely beautiful laser printer. The PCPI LaserImage 2000 was priced at the show at \$2,995; it has four different fonts and the ability to mix text and graphics. This printer emulates a Diablo 630; it

will work with any of the Zenith machines.

UCI, the makers of the EasyPC IBM emulator for the Z100, had the first issue of their *UCI Newsletter* on hand at the conference. The newsletter announced the anticipated arrival of three new UCI products for the Z100: EasyWIN, an "economical" hard-disk system; Easy-MATE, "two software drivers that enhance the storage capabilities of the EasyPC"; and I/O Plus, an IBM-emulation package. I/O Plus provides less IBM compatibility than their EasyPC, but it is also less than half the price.

Condor Computer Corporation was present to exhibit its new text editor, called "The Editor"—as well as to show Condor 3, the latest version of its popular data base management system for the '100 and IBM compatibles.

EWDP Software, Inc., introduced us to UTL. It's an enhancement to CP/M to simplify and improve the resident commands. Under UTL, for instance, both TYPE and the directory command allow bi-directional scanning of the display.

UTL was being offered as a special bonus to anyone who bought EWDP's data base manager, Filebase (for the '100 and IBM compatibles). The folks from EWDP were also proudly handing out copies of the positive reviews that Filebase has received in several noted

computer magazines.

Matt Gray of *Hilgraeve* was excited that several magazines had recently published reviews of HyperACCESS, his company's communications program. *InfoWorld* gave the program a four-disk (very good) rating in their July 14, 1986, issue, and the September 1986 *Software Digest* rated it first in a test of fifteen communications programs. As recently as October 28, 1986, *PC Magazine* had some very positive things to say about the program.

However, according to Gray, Hilgraeve is not just going to sit back and enjoy success. They are currently planning to release HyperACCESS 4.0. This new version will be available as an upgrade for current HyperACCESS users. It will not only be faster, but will also have additional file-transfer protocols, and will support modems that run up to 19,200 baud. The program will be able to learn any sequence of instructions you input, and then repeat it on command. Hilgraeve is not yet making any predictions as to the date of release for HyperACCESS 4.0.

Gray also mentioned that the current version of HyperACCESS would be available on 3½" disks. The 3½" format was released just after CHUGCON. With just a few keystrokes, users of the Z181 portable can transfer all or some of their files from the '181 to their desktop computers. Aren't things getting easy?

Celebrities and fanfare

On Saturday morning, Pat Gallagher gave a talk that was enough to make you never want to hold his job as Manager for Federal Sales at Zenith Data Systems. His talk, entitled "Zenith: Behind the Scenes in the Government Market," provided an account of what it's really like to go after a major government computer contract.

Two years ago, Gallagher said, four people worked in the federal sales department at Zenith. Now there are 54 people working in that office. In case you were wondering about this 1300% employee increase, Gallagher was quick to explain.

As you may have heard, Zenith recently won a big military contract for the Z248. It calls for Zenith to provide 90,000 '248s over a three-year period. (The '248 is based on the 80286 microprocessor, and it's compatible with the IBM PC AT. The big difference between it and the '241 is that the '248 runs faster—at 8 megahertz.)

According to Gallagher, the contract was won through a combination of exhaustingly hard work and a dedicated effort to exceed the minimum requirements of the government.

Gallagher indicated that the best way to win a big contract is to give the buyers exactly what they want. For the '248 sale, this meant designing brand-new

MicroSystems Call

Z-100 ↔ VAX ↔ Z-150/IBM PC

Communications Software

COMPLETE COMMUNICATIONS

- Unmatched VT-100 emulation including true 132-column, double-high, and double-wide characters.
- Error-free file transfers with XMODEM protocol.
- Communicate with VAX computers, other PCs, or information utilities such as The SourceSM, CompuServeTM, DELPHITM, NewsNetTM or MCI MailSM.

EASY-TO-USE

- Scripts, keyboard macros, and Hayes auto-dialing simplify your routine communications tasks.
- "Hot-key" leaves your DOS only a keystroke away.
- LotusTM-style menus are simple to learn and use, yet never get in your way.

VALUE

- Introductory price of \$49. For your Z-100, Z-150, Z-200, IBMTM PC, or compatible.
- Not copy-protected. 60-day guarantee.
- Send this coupon today!

YES! I want complete communications, send me copies of Call at the introductory price of \$49 each.

Please include \$1.50 for shipping, California residents add \$3.43 sales tax.

NAME:

ADDRESS:

CITY: STATE: ZIP:

Mail to: MicroSystems Company (408) 358-2224
16987 Frank Ave., Dept. 63, Los Gatos, CA 95030

Circle #101 on Reader Service Card

SIRIUS SOFTWARE INC.

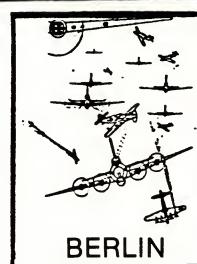
★ NOW AN AUTHORIZED ZENITH DEALER ★

We have all types of high quality Zenith computers.

Z-158's Z-200's Z-148's Z-181's

Please call for pricing in the configuration you need.

OUR OWN QUALITY SIRIUS SOFTWARE



SUPER TREK

A space war game in full color graphics. Do battle with dreaded Kligites. Avoid worm holes. Experience black holes.

\$29.95

KONGO

Avoid the obstacles and traps as you search for treasure in a variety of increasingly difficult mazes. Full color animated graphics.

\$29.95

BERLIN

Control over 100 planes at once while you take command of the air war over Europe during WW11. Play the American side or the German side against the computer. Berlin is for the Z-100 only. (one or more players)

\$39.95



Orders Only
Please PHONE
1-800-423-7801

Florida Residents
Inquiries
1-305-971-8068



SIRIUS SOFTWARE, INC., 1524 N.W. 65th Ave., Margate, FL 33063

Circle #212 on Reader Service Card

machines to adapt to the government's specific needs, and writing a series of different bids to fit every possible situation.

In all, about 50 or 60 people worked on the project—engineers and negotiators, writers and proofreaders. Most of them spent more than a few 20-hour workdays in an all-out effort to provide the perfect contract. At first, the crew wrote up 27 separate bids, but then worked these down to eleven.

Gallagher said that Zenith's goal was to provide "the highest technical quality at the lowest price . . . and we did it".

A historic moment

It was a proud moment for the Heath/Zenith community at Sunday night's banquet, when Heath made a presentation to the Smithsonian Institution. Bill Johnson (president of Heath Company) and Joe Schulte (president of Veritechology Electronics Corp.) presented an H8 and an H9 to John Ecklund, Curator of the Division of Physical Sciences for the Smithsonian. Ecklund is collecting computer memorabilia for an exhibit scheduled to open at the Smithsonian in the 1990s.

The H8 was the Heath Company's first digital computer. It came with 4K, but could be expanded to 64K. In addition, the H8 could be interfaced by way of the H9 terminal—that is, Johnson told us, "if you soldered all the keyboard connections together properly."

Since then, the microcomputer industry has come a long way. Home computers with 4K now seem inadequate, and most people would faint at the thought of soldering their keyboard connections. Perhaps most outrageous of all, however, is the fact that average people actually *own* home computers. Ecklund said it well when he accepted the two computers on Sunday: "We are so used to having micros that it doesn't seem strange to have a computer in the house; but I want to tell you, as a historian, that is a very strange thing. Very strange indeed."

One other contribution was made to the Smithsonian collection at Sunday night's banquet. Condor Computer Corporation donated an original copy of the first data base management system ever made for a microcomputer. Condor originally sold this program for \$10,000, but has since lowered the price to \$650 (yet another change for the better).

Winning prizes

Another exciting event at the banquet was the drawing for prizes. Door prizes donated by vendors had been awarded every hour during the conference, but the attendees of the banquet were eligible for even more prizes.

The *Sextant* table had two winners! Kieran O'Leary, *Sextant*'s retail sales manager and production assistant, won the first prize given out at the banquet: a

MS-DOS & CP/M CALLIGRAPHY II Printer/Graphics

 Our popular CALLIGRAPHY II program is now available for MS-DOS, as well as CP/M & CP/M-85. The CALLIGRAPHY II package includes a screen character editor, the GRAPHICS/TEXT FORMATTER and Font Set 1, a starter set of alphabets, pictures and borders. Available for EPSON and compatable printers. The price of the package including complete manual with tutorial, is \$69.00. (Calif. residents please add sales tax.)

 The GRAPHICS/TEXT FORMATTER program processes text files created with your favorite text editor or word processor to create documents with mixed text and graphics on your dot matrix printer. (This ad was produced using the GRAPHICS/TEXT FORMATTER.) It goes beyond most word processing systems in its ability to mix our custom fonts and graphics with standard text. Among the features are filling, centering and easy control of overlays and borders.

 Expand the library of fonts and graphics by ordering Font Sets 2, 3 and 4 at \$19.95 each. You can also create your own with the screen character editor.

SKYCASTLE

COMPUTER PRODUCTS
P.O. Box 1412, San Carlos, Ca. 94070

Phone (415) 254-3931

Circle #218 on Reader Service Card

VAX/PC Users

ZSTEMpc™-VT220 Emulator for the PC Series
High performance COLOR VT220. Double high/wide, 132 mode, smooth scrolling, downloadable fonts, user defined keys, 8-bit and full national/multinational modes, XMODEM and KERMIT, softkey/MACROS, DOS access \$150.00 Available soon: VT220 Style replacement keyboard.

ZSTEMpc™-VT100 Emulator for the PC Series
High Performance COLOR VT100 \$99.00

ZSTEMpc™-4014 Emulator for the PC Series
Use with ZSTEMpc-VT100, VT220, or stand alone. Interactive zoom and plan. Save/recall images to/from disk. Keypad, mouse, printer, and plotter support.

ZSTEM™-VT100 Emulator for Z-100s \$148.95
DECKHAND™ Utilities for the Z-100 & PC Series
MS-DOS Utilities with VAX/PDP-11 switch processing, DIR, COPY, DELETE, RENAME, TYPE with wildcards, full DATE processing, attribute processing, query, backup and more. \$ 49.00

Z-100 PC Memory Expansion with ZPAL™

Z-151/Z-161 Users

Use 256K bit memory chips on the original memory board. Extend the memory to 640K or 704K bytes with ZPAL-2 Decoder \$ 29.00

Z-158 Users

Extend your system to 768K bytes with 3 banks of 256K bit memory chips and ZPAL-158 decoder.

704K contiguous + 64K separate or

640K contiguous + 128K separate ZPAL-158 Decoder \$ 36.00

Z-138/Z-148 Users

Extend your system to 768K bytes with 3 banks of 256K bit memory chips and ZPAL-148 decoder.

704K contiguous + 64K separate or

640K contiguous + 128K separate ZPAL-148 Decoder \$ 36.00



Call today: VISA and MasterCard accepted
Quantity and dealer discounts available

30 day money back guarantee

AIR MAIL POSTAGE included for US & Canada

KEA Systems Ltd.

2150 West Broadway, Suite 412

Vancouver, B. C. CANADA V6K 4L9

Technical Support (604) 732-7411

TELEX 04-352848 VCR

Order desk (800) 663-8702 Toll Free

Trademarks IBM PC, Jr, XT, AT - IBM Corporation. VT100, VT220, DEC, VAX - Digital Equipment Corp. ZSTEMpc - KEA Systems Ltd.

Circle #175 on Reader Service Card



box of spare parts from Pair and a Spare. It appears to be a complete Heathkit, but we're still not exactly sure what the parts create when they are put together. I'm sure Kieran will let us know as soon as he finds out.

Beverly Voigt, *Sextant's* editorial and production assistant, won a project-planning software package called Super-Project Plus, from Computer Associates International, Inc. It will undoubtedly prove to be helpful in the office.

Unfortunately, another *Sextant* stalwart, Trish Pearson, did not win the HERO robot that was given away. Trish was hoping to win one so that her two children could learn about computers, robots, and sharing, all at once.

In addition to generous donations from many of the vendors, CHUG awarded two ZP-150s and a grand-prize H248 to lucky ticket-holders.

And working overtime

However, if we are going to talk about winners, good guy Bill Johnson, president of Heath Company, deserves the award for having the best sense of humor about himself. During the banquet Sunday night, Johnson told us a story about his son's reaction to Dad's hard work.

While in the second grade, Johnson's son came home from school one day and asked his mother why Dad always

brought so many papers home from the office. Mrs. Johnson patiently explained to her son that Dad worked very hard every day, and that sometimes, when he was not able to finish his work at the office, he had to bring it home in his briefcase and finish it after dinner.

At this point, Johnson's son looked up at his mother in wide-eyed innocence and asked, "Well, why don't they just put him in a slower group?"

Well, if it means anything to you, Bill, we're glad they didn't.

Suppliers Mentioned

Travis J. Barfield, Jr.
3320 Kensington St.
Virginia Beach, VA 23452

Condor Computer Corp.
1490 Eisenhower Place
Ann Arbor, MI 48104
313/971-8880

EWDP Software, Inc.
P.O. Box 40283C
Indianapolis, IN 46240
317/872-8799

Gemini Technology Corp.
13227 N.E. 20th St.
Bellevue, WA 98005
800/663-8331

Goldstein Software, Inc.
12520 Prosperity Drive, Suite 340
Silver Spring, MD 20904
301/622-9020

Hilgraeve, Inc.
P.O. Box 941
Monroe, MI 48161
313/243-0576

Lindley Systems
21 Hancock St.
Bedford, MA 01730
617/275-6821

P.C. Enterprises
P.O. Box 1269
Annandale, VA 22003
703/941-7991

Quic-n-Easi Products, Inc.
136 Granite Hill Court
Langhorne, PA 19047
215/968-5966

UCI Corp.
948 Cherry St.
Kent, OH 44240
216/673-5155; 800/824-2667

Weber & Sons, Inc.
3468 Highway 9, R.D. #4
Freehold, NJ 07728
800/225-0044; in New Jersey,
201/431-1128

\$749 20 Meg Z-100 Winchester

LINDLEY SYSTEMS announces a significant reduction in our hard drive pricing. Here are our new prices:

	Internal	External
10 Meg	\$ 699	\$ 899
20 Meg	\$ 749	\$ 949

These systems include:

- The Konan DGC-2000 Controller
- A Seagate Winchester drive
- Our Software Driver
- Appropriate Cables

DGC-2000 controller and software, \$350
software only, \$60

KONAN Controller

The Konan controller board is a single slot S-100 card which will connect with one or two ST-506 compatible hard drives. Because this board contains the latest technology interface, it uses much less space and power than previous controllers.

Note that with the Konan board, there are no traces to cut, no ROMs to replace. Just plug in the board and go. Whether you decide to install the drive

internal or external, the only tool you need is a screwdriver. (Internally mounted drive units simply replace one of your half-height floppy drives.)

WINCHESTER DRIVE

Our Winchester systems include either the ST-213 (10 Megabyte) or ST-225 (20 Megabyte) drive units from Seagate.

DEVICE DRIVER

Our new device driver supports drive sizes up to 256 Megabytes, with up to 8 partitions, each up to 32 Megabytes. You can use one or two drives (up to 512 Mb on line!) The driver operates under both MS-DOS 2 and 3.

Our hard disk systems are shipped with two versions of the driver software: one for native Z-100 mode, and one for use with the Gemini PC emulator. Any partition can be mounted either under Z-100 mode, Gemini mode, or both. Thus, you can put Z-100 programs in one partition; Gemini programs in another; and your data, shared between both modes, in a third.

LARGE CAPACITY

Please call for pricing and availability of large capacity drives. We are seeking (no pun intended) the best value!

LINDLEY SYSTEMS 21 Hancock Street Bedford, MA 01730
(617) 275-6821

Circle #118 on Reader Service Card

--- SHIPPING NOW --- SHIPPING NOW --- SHIPPING NOW ---

WILDFIRE

An
**INCREDIBLE PERFORMANCE
IMPROVEMENT**
from
Software Wizardry

SPEED... WILDFIRE has spirit. It will spur your Z-151 or Z-161* to run faster than the 8MHz Z-158! And WILDFIRE isn't just a simple crystal change like some products; it gives you more speed and much greater compatibility.

COMPATIBILITY... WILDFIRE uses the NEC V-20 chip plus lots of other enhancements to make an unbeatable combination of performance and reliability. It outdoes the 80286 processor chip additions for compatibility with the IBM PC and costs up to \$600.00 less!

FEATURES... the heart of WILDFIRE is a daughter board that mounts on your processor board, saddled comfortably between it and the video board. Several higher-speed chips are included to replace socketed chips on your processor board. A high speed/low speed switch that mounts on your machine's front panel keeps WILDFIRE on a tight rein, under your control to trot at normal speed, or break into a real gallop when you really want to ride!

PRICE... Software Wizardry brings you WILDFIRE, complete with attractive front panel, installation instructions, and reliable daughter-board design, for only \$249!

*** 15-DAY MONEY-BACK GUARANTEE! ***

If you order now, you can try WILDFIRE with a 15-day money-back guarantee! If you don't like it, you can return it within 15 days and get a full refund!

WILDFIRE...the ultimate solution to your speed problems. And with our special low price, AND our 15-day money-back guarantee, you can't go wrong! Order now!

Contact your local dealer, or order direct from Software Wizardry by calling (314)724-1738. Dealer inquiries invited.

1106 First Capitol Drive
St. Charles, MO 63301
(314)724-1738



Classified Ads

Hardware

H89A, 64K, two drives, software, manuals, etc. Also, Epson MX-80A. \$475. Vincent C. Bartolo, 213/516-6774.

H89, 64K, 5 1/4" (H17) drive, dual 8" (H47) drives, manuals, top condition, \$700. 716/394-7316.

Software

PRINTER CONTROL. "COM" programs to control your printer from

keyboard. MS-DOS/CP/M versions. Tell me printer, operating system. \$25. CUSTOM PRINTER FONT(S) for printers that support "downloaded" characters. No program over \$25. Visa and MC. Custom Computer Service, 503/760-4690, 7466 S.E. 112th, Portland, OR 97266.

CROSS ASSEMBLER FOR 6502. Runs on IBM PC or compatible under MS-DOS. Produces absolute Intel code file and listing file. \$35 postpaid U.S.A. Neauware, P.O. Box 143, Saunderstown, Rhode Island 02874.

Information

PERSONAL COMPUTER OWNERS can earn \$1,000 to \$5,000 monthly selling simple services performed by their computer. Work at home—in spare time. Get free list of 100 best services to offer. Write: C.I.L.D.H., P.O. Box 60369, San Diego, CA 92106-8369.

Coming Up in Sextant

Sextant's fifth anniversary issue. . .

we'll look at the vendors, authors, and subscribers who've helped keep the *Sextant* community strong for the past five years.

How to Order a Classified Ad

To get in touch with your fellow Heath/Zenith users, place your short notice in *Sextant's* classified section. The rate is only 75¢ per word, with a minimum of 15 words.

Please omit all *specific* references to software, unless the package is unopened.

Send your typewritten ad and pay-

ment to: *Sextant*, Classified Ad Department, 716 E Street, S.E., Washington, DC 20003. Please include your name and phone number for our records.

Deadline is January 21 for the March-April issue, and March 19 for the May-June issue.

IMAGE ACE II Video Capture System



\$295.00
complete

For PC Compatible Computer Systems

- single slot card
- software included
- complete manual
- video cable

Digitize video images from cameras, television tuners, and video recorders directly to your PC compatible display

- No additional video monitor needed
- 320 x 200 x 4 levels B/W or pseudocolor
- 1.3 second capture time
- Works with IBM-PC/XT/AT or PC compatible with standard color graphics adapter.
- Software included to capture and save images in DOODLER-V PIC format
- Includes stand-alone software to capture, edit, save, and print digitized images.*
- Transfer captured images to Z-100 using DOODLER-V

Full 90 day manufacturers warranty

IMAGE ACE II Video Capture System manufactured by Lodge Electronics

Available from . . .



Software Graphics Tools
3620 Amazon Drive

New Port Richey, FL 33553

C.O.D. Orders accepted . . . Call 813-376-5457

* Software provided by Lodge Electronics is written in BASIC, and includes source code. User modifications for specific applications are easily done with GW-BASIC.

Circle #172 on Reader Service Card

H89 20-MEGABYTE WINCHESTER Only \$595

TIRED of swapping floppies and living with limited space?

- Boot from one 20-meg or two logical 10-meg drives
- Uses left-side memory slot
- Seagate ST225 drive and Western Digital controller can also be used in H/Z150-160
- External option, \$125; printer port, \$10
- Drive and controller for H/Z150-160, \$400

H89 & H150 SPEED DOUBLER

Stop WAITING! Increase your computer's speed NOW!
Easily installed. NO trace cuts or permanent mods.

H89

- Software-select 2- or 4-MHz
- Supports Heath, Magnolia, CDR CP/M, and HDOS
- 4-MHz Z80A included
- Assembled and tested \$34.95
- Complete kit with s/w \$24.95

- Satisfaction guaranteed!
- Runs at 6.77 or 4.77 MHz
- Hardware reset included FREE
- 2.5 Norton Index with V20
- Assembled and tested \$34.95
- Complete kit \$24.95

EXPANDED MEMORY for H/Z150-160

SUPER PAL

1.2 meg of memory. Disk driver and LIM expanded memory driver included. \$49.95

MT PAL

640K or 704K bytes on standard memory card (2 banks 256K and 2 or 3 banks 64K) \$19.95

ORDER NOW! by writing or calling:

Micronics Technology
(904) 897-4257

449 Barbados Way, Niceville, FL 32578

We accept checks, Visa, MasterCard.

Shipping and handling: \$3

Winchester, \$15

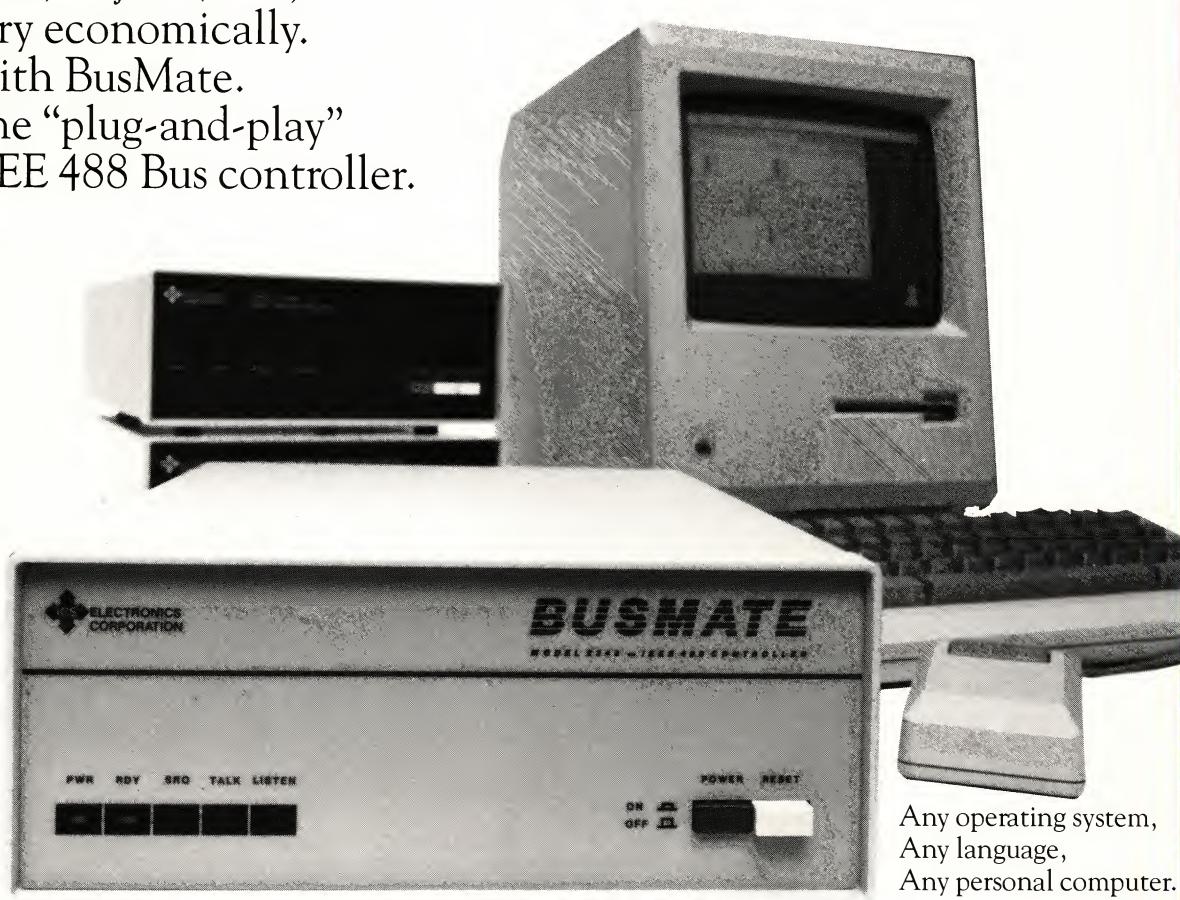
PAL is a registered trademark of Monolithic Memories, Inc.

Circle #207 on Reader Service Card

BusMateTM

Establish and control the IEEE 488 Bus from any personal computer. More quickly and easily than you thought possible.

And, at just \$695, very economically. With BusMate. The "plug-and-play" IEEE 488 Bus controller.



Any operating system,
Any language,
Any personal computer.

BusMate brings the full power of the IEEE 488 Bus to your personal computer. BusMate establishes the bus—which you may know also as GPIB, or HP-IB—and turns your PC into a full-featured 488 bus controller.

Your PC can be an Apple Macintosh, IIc, or IIe...or an IBM PC or compatible like the Zenith or Compaq...an S-100 system...or just about any PC you can name.

Plug BusMate into the RS-232C serial-data connector on the back panel of your computer and it's ready to go. Self-powered and self-contained, BusMate has no system configuration requirements, requires no third-party or other special software, and no floppy disks.

Call or write for complete BusMate information.

Circle #115 on Reader Service Card

BusMate
From ICS...the leader
in IEEE 488 Bus controllers
and interface instrumentation.



ICS Electronics Corporation
2185 Old Oakland Road
San Jose, CA 95131
(408) 263-4844

Special Fifth Anniversary Offer

This issue of Sextant marks the completion of our first five years serving the Heath/Zenith community. In our next issue (March-April 1987), we'll have a special section looking back at the past five years.

We'd like to celebrate our anniversary by offering you a chance to extend your subscription at the lowest rate

we've ever offered.

Here's how it compares: If you'd purchased the first five years of Sextant at a retail store, you would have paid \$76.70. By subscription, these first 26 issues would have cost you \$64.82. Now, in celebration of our fifth anniversary, you can add 26 issues of Sextant to your subscription for only \$55.55.

Take advantage of this special offer to extend your subscription—use the reply card bound into this issue between pages 14 and 15. Do it today!

26 issues in 5 years?

Sextant began publication in March of 1982 as a quarterly magazine. Four issues were published in 1982 and 1983. Frequency was increased to six issues per year for 1984, 1985, and 1986. (The last issue is mailed late in December, with a cover date of the following year.)

Thus a total of eight issues were published in Sextant's first two years, with 18 more in the next three years.

We'd like to be able to publish Sextant more often next year. If that happens, the next 26 issues could come out within 3½ years.